NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DIRECT-SEQUENCE SPREAD-SPECTRUM ACOUSTIC COMMUNICATIONS WITH CRV DECOMPOSITION

by

Pavlos Angelopoulos

June 2004

Thesis Co-Advisor:	Roberto Cristi
Thesis Co-Advisor:	Kevin B. Smith

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Direct-Sequence Spread-Spectrum Acoustic Communications with CRV Decomposition			**5. FUNDING NUMBERS**
6. AUTHOR(S) Pavlos Angelopoulos			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Direct-Sequence Spread-Spectrum (DS-SS) is among the preferred modulation techniques for military applications. DS-SS offers three greatly desired characteristics. It allows for the development of Low Probability of Detection (LPD) and Low Probability of Intercept (LPI) systems and has a very good performance in fading channels. This thesis investigates the performance of the "Cross-Product RV (CRV) decomposition" as the basis of blind-equalization algorithms. The CRV is a rank-revealing decomposition alternative to the Eigenvalue Decomposition (EVD) that can provide a recursively updated estimate of the signal and noise subspace at a reduced computational cost. The CRV updating algorithm is implemented in MATLAB and evaluated in a previously proposed communication scheme intended for use in an underwater acoustic network called Seaweb. The underwater channel is modeled with the Monterey-Miami Parabolic Equation Model (MMPE) for various multipath perturbations. The receiver performance is examined using Monte Carlo simulation. Bit-error rates versus signal-to–noise ratio are presented for various, noise assumptions, and receiver synchronization assumptions.

14. SUBJECT TERMS: Underwater acoustic communications, Seaweb, Reverberation, TL, Ambient noise, Interface roughness, Internal waves, Turbulence, Volume perturbations, Sound speed perturbations, Doppler, MMPE, Direct Sequence Spread-Spectrum, Gold-codes, CRV decomposition, Subspace decomposition, Blind equalization, Matched filter	**15. NUMBER OF PAGES** 115
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

DIRECT-SEQUENCE SPREAD-SPECTRUM ACOUSTIC COMMUNICATIONS WITH CRV DECOMPOSITION

Pavlos Angelopoulos
Lieutenant, Hellenic Navy
B.S., Hellenic Naval Academy, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
and
MASTER OF SCIENCE IN ENGINEERING ACOUSTICS

from the

NAVAL POSTGRADUATE SCHOOL
June 2004

Author: Pavlos Angelopoulos

Approved by: Roberto Cristi
Thesis Co-Advisor

Kevin B. Smith
Thesis Co-Advisor

John Powers
Chairman, Department of Electrical
and Computer Engineering

Kevin B. Smith
Chairman, Engineering Acoustics Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Direct-Sequence Spread-Spectrum (DS-SS) is among the preferred modulation techniques for military applications. DS-SS offers three greatly desired characteristics. It allows for the development of Low Probability of Detection (LPD) and Low Probability of Intercept (LPI) systems and has a very good performance in fading channels. This thesis investigates the performance of the "Cross-Product RV (CRV) decomposition" as the basis of blind-equalization algorithms. The CRV is a rank-revealing decomposition alternative to the Eigenvalue Decomposition (EVD) that can provide a recursively updated estimate of the signal and noise subspace at a reduced computational cost. The CRV updating algorithm is implemented in MATLAB and evaluated in a previously proposed communication scheme intended for use in an underwater acoustic network called Seaweb. The underwater channel is modeled with the Monterey-Miami Parabolic Equation Model (MMPE) for various multipath perturbations. The receiver performance is examined using a Monte Carlo simulation. Bit-error rates versus signal-to–noise ratio are presented for various, noise assumptions, and receiver synchronization assumptions.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I am very grateful to my beautiful wife Natassa and my children Angelo and Maira for their loving support and the many sacrifices they made during my studies at the Naval Postgraduate School. Without their help and motivation none of this would have been possible.

I would also like to thank my advisors, Prof. Roberto Cristi and Prof. Kevin B. Smith for their patience and valuable guidance throughout the whole process of this project.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Underwater wireless networks have many military and commercial applications. With such networks, the control and coordination of various assets, such as mini-submarines, divers, submerged instruments and Unmanned Underwater Vehicles (UUVs) is possible without the burden of cables or human intervention.

The underwater environment is possibly the most challenging environment for wireless communications. The propagation of sound is greatly affected by transmission loss (TL), ambient noise, reverberation, and the overall variability of the channel. These limiting factors impose various trade-offs in the design of a reliable communication system. A higher transmission rate implies greater bandwidth, higher TL and intersymbol interference (ISI) but an almost negligible channel variation over a single-bit interval.

Direct-sequence spread-spectrum (DS-SS) is one of the preferred communication techniques in military applications. Spread-spectrum modulation has a very good performance in multipath environments and allows for the development of Low Probability of Detection (LPD) and Low Probability of Intercept (LPI) systems. Because of its desired characteristics, DS-SS is well suited for shallow-water communications.

This thesis investigates the performance of DS-SS with blind-equalization in simulated ocean channels. The equalization algorithms examined are based on the "Cross-Product RV (CRV)," a relatively new subspace decomposition algorithm proposed as an alternative to the Eigenvalue decomposition (EVD). The CRV updating algorithm provides a recursively updated estimate of the signal and noise subspace at less computational cost than the EVD. The transmitter and receiver structures are implemented in MATLAB and the channel impulse response is simulated with the Monterey-Miami Parabolic Equation model (MMPE). The impulse responses extracted account for various channel perturbations (interface roughness, internal waves, etc.) and also for the Doppler effect caused by the source motion.

The performance of the proposed communication scheme is evaluated using Monte Carlo simulation for various signal-to-noise ratios (SNR) and design parameters

(packet size, data rate and channel length). For all the individual perturbations simulated with the MMPE model a Bit Error Rate (BER) below 10^{-5} is achievable for a relatively low SNR. In the case where all perturbations are combined or synchronization between the receiver and the transmitter is not achieved, the receiver's performance is limited.

I. INTRODUCTION

A. BACKGROUND

The primary purpose of this thesis is to improve upon two previously proposed Direct-Sequence Spread-Spectrum (DS-SS) communication schemes [1, 2] intended for use in an underwater acoustic communication network known as Seaweb. Seaweb is being developed by the Space and Naval Warfare Systems Center, San Diego, [3] and is motivated by a requirement for wide-area undersea surveillance in littoral waters. This undersea wireless network provides the command, control and communications infrastructure [4] for coordinating appropriate assets, such as mini-submarines, divers and Unmanned Underwater Vehicles (UUVs) to accomplish a given mission in an arbitrary ocean environment. The first underwater communication scheme [1] involved a Direct-Sequence Differential Binary Phase-Shift Keying with quadrature spreading (DS-IQ-DBSK). Error-correction coding and a RAKE receiver for multipath reception were used to improve system performance. The second and latest communication scheme as part of a previous thesis research [2] tried to develop an improved receiver structure based on blind-equalization algorithms. In both schemes [1, 2] the impulse response of the underwater acoustic channel was generated using the *Bellhop* acoustic propagation model [5]. The *Bellhop* model is a "variant of Gaussian beam tracing and is especially suited for high-frequency deep-water problems where normal mode and parabolic equation (PE) models are not practical [6]". In this thesis, the channel impulse response is estimated using the Monterey-Miami Parabolic Equation (MMPE) model described in [7]. This is a range dependent wave-theory model based upon the parabolic approximation of the wave equation [6]. It is a widely used numerical tool for solving the acoustic wave equation especially for low frequencies and/or range-dependent environments where it is computationally more efficient.

B. SEAWEB UNDERWATER NETWORK REQUIREMENTS

Like all other military applications, Seaweb must ensure communication security in two ways. It must limit the ability of unauthorized users to detect the presence of the communication signal (i.e., low probability of detection) and at the same time decrease the ability of a hostile observer to "listen in" when communications are taking place (i.e.,

low probability of intercept). The aforementioned requirements are met by using DS-SS as the modulation scheme for packet transmission. DS spread-spectrum systems are able to hide the signal in the background noise by operating at low bit-energy per noise-spectral-density (E_b/N_o). Also due to the pseudo-random properties of the code (chipping sequence) used to spread the message signal, DS systems have a very low probability of intercept. The message can be demodulated only by the intended receivers who know the exact replica of the code used at the transmitter [8].

Other Seaweb requirements include utility packets of fixed length (72 bits) and a limited operating bandwidth of 5 kHz (9–14 kHz operating frequencies). The desired information transmission rate is 40 to 100 bits per second (bps) at a bit-error rate (BER) of 10^{-5}. The communication range is between three to five kilometers and the operating depth is on the order of 50 to 200 m.

C. GOALS AND METHODOLOGY

The first step in improving the previously proposed signaling schemes is to use a more accurate underwater acoustic model to account for the temporal and spatial variability of the channel. As already mentioned, this model is the MMPE model and will be described in more detail in Chapter III of this thesis. For the sake of comparison various impulse responses are extracted corresponding to different undersea environments.

The combination of the DS-SS transmitting scheme and the receiver structure based on blind equalization algorithms was proven very robust as part of a previous thesis [2]. However, the algorithm worked completely offline and the whole received packet was needed in order to calculate the filter coefficients and to demodulate the transmitted signal efficiently. As part of this thesis research a relatively new subspace tracking algorithm, the cross product RV, also known as CRV [9], is used to allow for a real-time estimation of the desired subspaces. The transceiver with the CRV algorithm is implemented in MATLAB for various combinations of multipath, white and colored noise.

D. BENEFITS OF STUDY

Underwater wireless communications are much desired in military applications. The possibility to maintain signal transmission without the burden of cables enables the operation of underwater robots and vehicles (UUVs) and the gathering of data from sub-

merged instruments without human intervention. An underwater mobile network such as Seaweb has many military and commercial applications (submarine communications, ocean exploration, etc.).

E. THESIS ORGANIZATION

The remainder of this thesis is organized into five chapters. Chapter II discusses the general characteristics of underwater acoustic communication channels. Chapter III provides a description of the Monterey-Miami Parabolic Equation (MMPE) model and an outline of the theory behind the simulated perturbations. Chapter IV describes the CRV algorithm, proposed as a substitute of the Eigenvalue Decomposition (EVD) used in [2]. The CRV updating algorithm can provide a recursively updated estimate of the signal and noise subspace. Chapter V reviews the previously proposed signaling scheme [2] and presents the new scheme based on the CRV algorithm. The performance of the new transmitter/receiver structure is evaluated under different channel conditions and design parameters. Finally, Chapter VI reviews and summarizes the important results and recommends follow-on work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. CHARACTERISTICS OF UNDERWATER ACOUSTIC COMMUNICATION CHANNELS

The sea forms a remarkably complex waveguide medium for the propagation of sound. The purpose of this chapter is to describe the dominant factors affecting sound propagation in shallow water and the difficulties those factors impose on designing a reliable communication system. The observation of sound waves is greatly affected by transmission loss (TL), ambient noise, and the temporal and spatial variability of the channel. Transmission loss and noise determine the signal-to-noise ratio (SNR), the maximum transmission range, and the available communication bandwidth. The temporal and spatial variability of the channel induced intersymbol interference (ISI), which influences system design (i.e., the choice of the modulation/signal detection method and the transmitter/receiver structure).

A. TRANSMISSION LOSS (TL)

The weakening of an acoustic signal as it travels through the sea is quantitatively described by *transmission loss,* TL. According to [10] and [11], TL is defined as the ratio in decibels between the acoustic intensity $I(r)$ in W/m^2 at a field point (of distance r) and the intensity $I_0(r_0)$ measured at a distance $r_0 = 1$ m from the sound source (transmitter),

$$\text{TL [dB ref. } r_0] = -10\log\left[\frac{I(r)}{I_0}\right] = 10\log\left[\frac{I_0}{I(r)}\right]. \tag{2.1}$$

For both plane and spherical waves, the acoustic intensity is proportional to the square of the pressure amplitude; therefore, TL can also be expressed as

$$\text{TL [dB ref. } r_0] = -20\log\left[\frac{P(r)}{P_0}\right] = 20\log\left[\frac{P_0}{P(r)}\right]. \tag{2.2}$$

Transmission loss can be decomposed into two terms, $\text{TL}_{\text{Spreading}}$, caused by energy spreading and $\text{TL}_{\text{Attenuation}}$, due to sound attenuation. Energy spreading depends on the propagation distance and the underwater environment characteristics. For the case of an omni-directional sound source in an unbounded medium, sound energy undergoes

spherical spreading and TL due to spreading is proportional to the square of the transmission range,

$$\mathrm{TL}_{\text{Unbounded, Isospeed}}\ [\text{dB ref. 1 m}] = 20\log r. \tag{2.3}$$

In isospeed underwater channels where the medium is bounded by two plane and parallel surfaces, the TL in the far-field is proportional to the first power of the propagation distance (cylindrical spreading),

$$\mathrm{TL}_{\text{Far-field, Bounded, Isospeed}}\ [\text{dB ref. 1 m}] = 10\log r. \tag{2.4}$$

In shallow water channels acoustic energy undergoes both types of spreading mentioned above–spherical spreading close to the source and cylindrical spreading at great distances [12]. Thus a more practical expression for the TL in shallow water is given by [10]

$$\mathrm{TL}_{\text{Shallow Water Spreading}}\ [\text{dB ref. 1 m}] = \begin{cases} 20\log r & r \le r_t \\ 10\log r + 10\log r_t & r > r_t \end{cases}. \tag{2.5}$$

The transition range r_t (i.e., the range at which the change from spherical to cylindrical spreading occurs) for the case of an isospeed shallow water channel with fast bottom, is estimated by [10]

$$r_t = \frac{H}{2\theta_c}, \tag{2.6}$$

where H is the bottom depth in meters and θ_c is the critical angle in degrees.

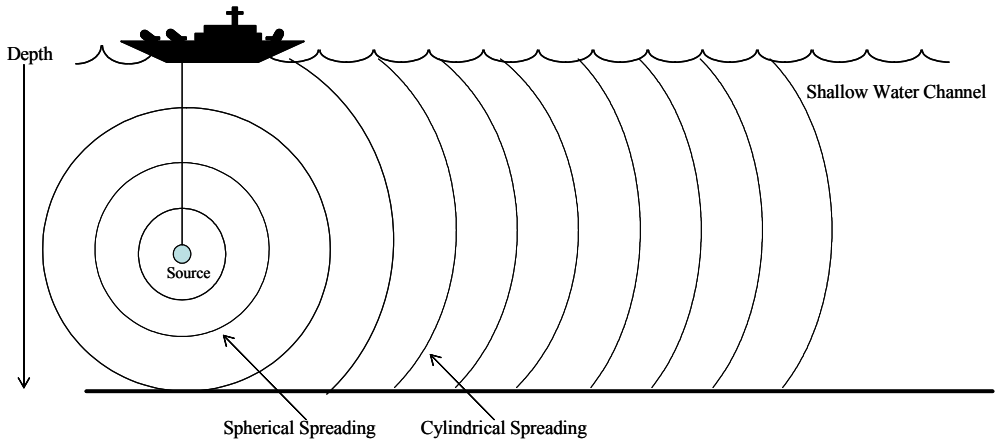

Figure 1. Spreading of Acoustic Energy in a Shallow Water Channel.

Attenuation of the field is the effect of sound absorption. Absorption involves the process of converting acoustic energy into heat, thus resulting in a true loss of energy. The dominant causes of absorption in seawater are the ionic relaxation of magnesium sulfate ($MgSO_4$), the boric acid (H_3BO_3) ionization, and the shear viscosity of seawater [10, 11]. Sound absorption is quantitatively described by the absorption coefficient α with units of dB/m. One of the most accurate expressions for the frequency dependent absorption coefficient is provided by Fisher and Simmons [10] and is based on experimental data;

$$\alpha\ [\text{dB/m}] = \left(\frac{A}{f_1^2 + f^2} + \frac{B}{f_2^2 + f^2} + C \right) f^2. \tag{2.7}$$

The terms A, B, C, depend on temperature and hydrostatic pressure whereas the relaxation frequencies f_1 and f_2 are functions of temperature only. Specifically, after the approximation given in [10]

$$f_1\ [\text{Hz}] = 780 e^{T/29}, \tag{2.8}$$

$$f_2\ [\text{Hz}] = 42000 e^{T/18}, \tag{2.9}$$

$$A = 8.3 \times 10^{-5} (S/35) e^{[T/31 - Z/9 + 1.8(pH-8)]}, \tag{2.10}$$

$$B = 0.022 (S/35) e^{(T/14 - Z/6)}, \tag{2.11}$$

and

$$C = 4.9 \times 10^{-13} e^{(-T/26 - Z/25)}, \tag{2.12}$$

where Z is the water depth in km, T is the temperature in °C, S is the salinity in ppt and pH is the acidity. The absorption coefficient in seawater and frequencies between 100 Hz and 1 MHz is shown in Figure 2 (from Equation (2.7) with parameter values given in the figure caption). In shallow water (Z < 150 m), depth, temperature and salinity have little effect on the absorption of sound.

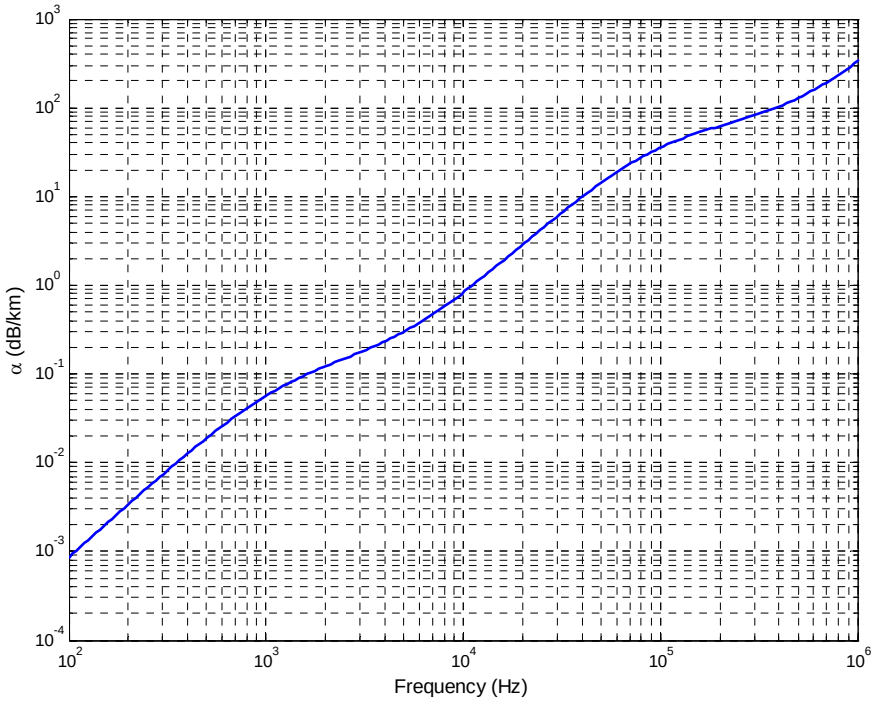

Figure 2. Absorption Coefficient (dB/km) in Seawater (pH = 8, S = 35 ppt, Z = 0.1 km, T = 15°C).

Now that simple forms for energy spreading and sound absorption are defined, a general approximation of the transmission loss in shallow water is given by [9, 11]

$$\text{TL [dB ref. 1m]} = \text{TL}_{\text{Spreading}} + \text{TL}_{\text{Absorption}} = \begin{cases} 20\log r + ar & r \le r_t \\ 10\log r + 10\log r_t + ar & r > r_t \end{cases}, \quad (2.13)$$

where the transmission range r and the transition range r_t are in meters and the absorption coefficient α in dB/m. The range and frequency dependency of TL for a fast bottom environment is illustrated in Figure 3 (for the parameters given in the captions of Figures 2 and 3). The frequencies chosen to evaluate Equation (2.13) are the lowest and highest operating frequencies of Seaweb, f_1 = 9 kHz and f_2 = 14 kHz, respectively. A third frequency (f_3 = 50 kHz) much higher than f_2 is used to demonstrate the logarithmic increase of the TL at high frequencies. In Figure 4, the TL is plotted as a function of frequency for the maximum transmission range (5 km) according to Seaweb specifications.

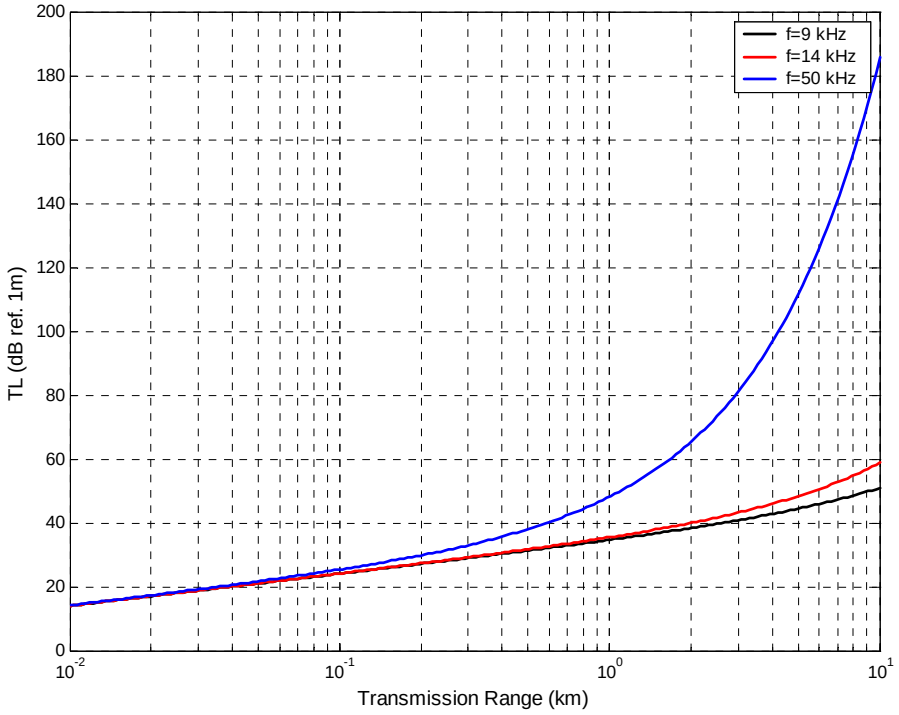

Figure 3. Range and Frequency Dependency of TL $\left(H = 150\,\text{m},\ \theta_c = 30^\circ\right)$.

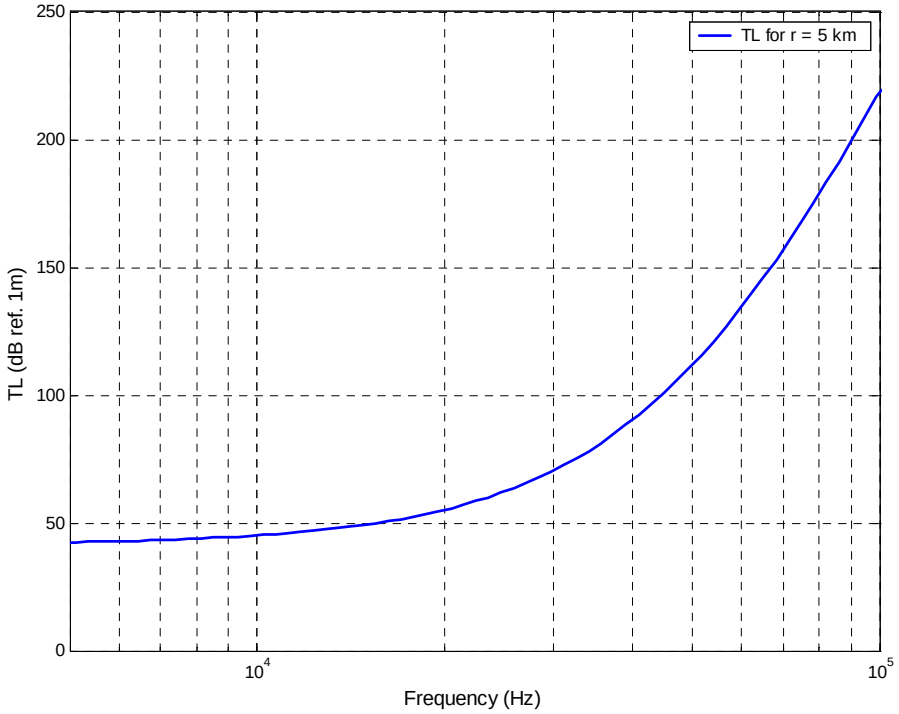

Figure 4. TL for Frequencies 5 kHz – 100 kHz $\left(H = 150\,\text{m},\ \theta_c = 30^\circ\right)$.

As seen in Figure 4, for a frequency of 12 kHz (the mean frequency of Seaweb), the TL is approximately 45 dB; whereas for a frequency of 50 kHz the TL increases logarithmically assuming the value of 110 dB. The increase of the TL at high frequencies is the reason the available communication bandwidth of Seaweb is limited to 5 kHz. Underwater communication channels are severely band limited compared to the channels used in RF communications. Because of the limited bandwidth, it is very difficult to achieve a reliable link at high bit rates. By increasing the data rate, the period of the message bit decreases and the bandwidth of the signal increases. For the sake of comparison, each channel of the 802.11g wireless network according to the IEEE standard has a bandwidth of 25 MHz and a maximum throughput of 54 Mbps, whereas an underwater network can only have a few kHz of bandwidth and a much lower data rate.

B. AMBIENT NOISE

The underwater environment everywhere contains ambient noise. Ambient noise is the total noise background minus the noise generated by the transmitting equipment used in a certain application. Noise observed in the ocean exhibits strong frequency dependence as well as geographical dependence. In deep water, there are many sources of ambient noise and each source dominates in a different frequency band. At low frequencies (below 20 Hz) the main noise sources are the ocean turbulence and the various tectonic processes such as earthquakes and volcanic eruptions [13]. In the frequency band 50 to 500 Hz, distant ship traffic and distant storms are the main sources of noise [12]. Between 0.5 and 50 kHz, the detected noise is associated with the breaking of the sea surface and bubble formations. The state of the local sea surface is weather dependent and closely connected with the prevailing wind speed. The greater the wind speed the greater the ambient noise contribution. Above 50 kHz, thermal noise due to the molecular agitation of the water molecules predominates [10]. The interested reader can read more about ambient noise and its sources in References [14] and [15]. The ambient noise spectrum level (ANL) due to the combined effects of the aforementioned sources is illustrated in Figure 5. In the frequency band (9 to 14 kHz), at which Seaweb operates, the wind speed (or sea state) is the main source of noise. For example, at 12 kHz and a wind speed of 15 knots, the ambient noise spectrum level is approximately 45 dB ref. 1 μPa. In general, the

ANL decreases when the frequency increases, and falls at a rate of 15 dB/decade when the frequency range is between 0.5 and 50 kHz.

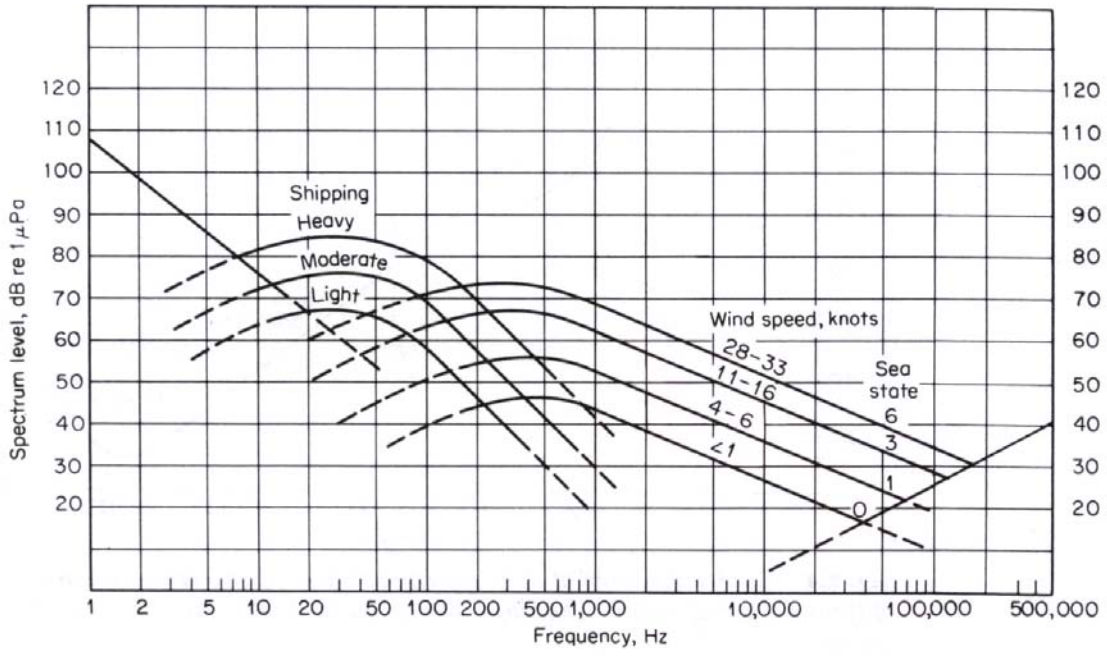

Figure 5. Average Deep-Water Ambient Noise Spectrum Level [From Ref. 12].

Wenz curves [15] (or modified ones like Figure 5) are commonly used to estimate the ambient noise spectrum level under various conditions (e.g., shipping density, sea state, etc.). The following expressions [16] are a quantitative description of Wenz curves:

$$\mathrm{ANL}_{\mathrm{Turbulence}} = 17 - 30\log(f), \tag{2.14}$$

$$\mathrm{ANL}_{\mathrm{Shipping}} = 40 + 20(D - 0.5) + 26\log(f) - 60\log(f + 0.03), \tag{2.15}$$

$$\mathrm{ANL}_{\mathrm{Wind}} = 50 + 7.5\left(w^{0.5}\right) + 20\log(f) - 40\log(f + 0.4), \tag{2.16}$$

$$\mathrm{ANL}_{\mathrm{Thermal}} = -15 + 20\log(f), \tag{2.17}$$

and

$$\mathrm{ANL}_{\mathrm{Deep\ Water}} = 10\log\left(10^{\frac{ANL_{Turbulence}}{10}} + 10^{\frac{ANL_{Shipping}}{10}} + 10^{\frac{ANL_{Wind}}{10}} + 10^{\frac{ANL_{Thermal}}{10}}\right), \tag{2.18}$$

where D is the shipping density with values between 0 and 1 (1 for heavy traffic), w is the wind speed in m/s, f is the frequency in kHz, and $\mathrm{ANL}_{\mathrm{Deep\ Water}}$ is the ambient noise level

in deep water with units of dB ref. 1 μPa. Equation (2.18) is evaluated for various shipping densities in Figure 6 and for various wind speeds in Figure 7.

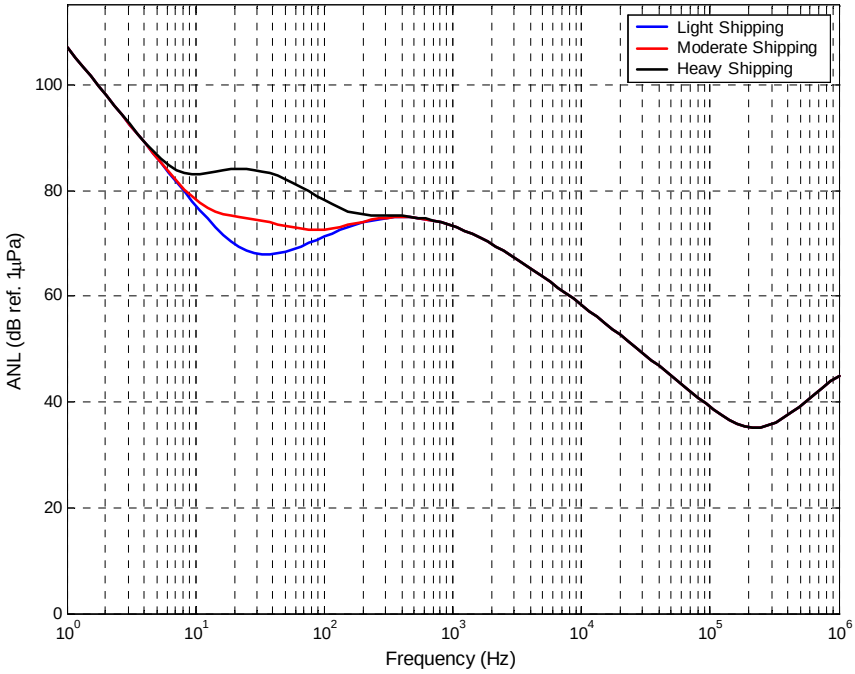

Figure 6. Ambient Noise Level in Deep-water for Wind Speed w = 15 m/s.

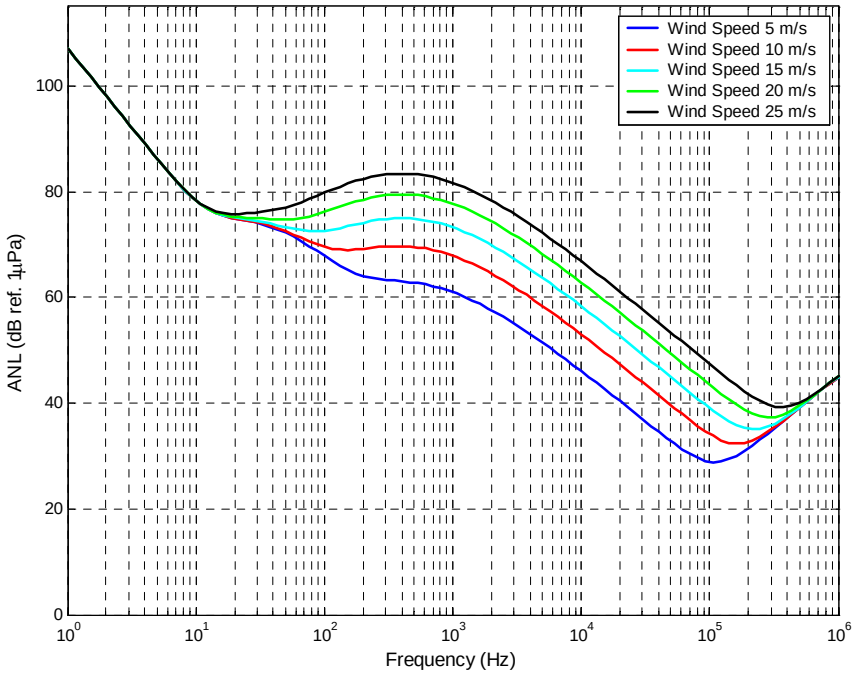

Figure 7. Ambient Noise Level in Deep-water for Moderate Shipping (D = 0.5).

In shallow water, ambient noise levels are very difficult to define because two additional noise sources, man-made noise and biological noise (dolphins, snapping shrimp, etc.) exist. These sources vary greatly both in time and location. Man-made noise is higher close to harbors and coastal areas with industrial activity. Noise from biologics is in general unpredictable. It is quite difficult to estimate beforehand and existing measurements (if any) are only valid in the specific location they were taken. However, in the absence of industrial and biological noise, the ambient noise level depends mostly on the strength of locally generated wind and is about 5 dB higher than the deep water levels for the same frequency [15].

Both transmission loss and the ambient ocean noise contribute to the degradation of the acoustic signal as it propagates away from the source. By combining Equations (2.13) and (2.18) and assuming that the noise generated by Seaweb equipment is zero, we can calculate the signal-to-noise ratio (SNR) of the underwater communication channel from

$$\text{SNR [dB]} = \text{SL} + \text{DI}_\text{T} + \text{DI}_\text{R} - \text{TL} - \text{ANL}, \tag{2.19}$$

where SL is the source level in dB ref. 1 μPa, 1m, and DI_T, DI_R are the directivity indices of the transmitter and the receiver, respectively. The directivity index is a quantitative measure of the focusing of acoustic energy, and for an omni-directional source or receiver, $\text{DI} = 0$. The SL of the low frequency omni-directional "telesonar" transducer AT-408 (a candidate for Seaweb) according to the OEM specifications [17] is 180 dB ref. 1 μPa, 1m at full power. The resulting SNR of the channel is illustrated in Figure 8. We can see from this figure that for a frequency f = 14 kHz and a transmission range of r = 5 km, the SNR is approximately 76 dB. Needless to say, this is only a rough and rather optimistic estimate of the SNR found in a real underwater environment. The frequency dependent SNR for various transmission ranges is shown in Figure 9. The reason the frequency band of 9 to 14 kHz is chosen for the Seaweb network is clear since this band accounts for the flat portion (SNR is almost constant) of the curves corresponding to r = 3 and r = 5 km.

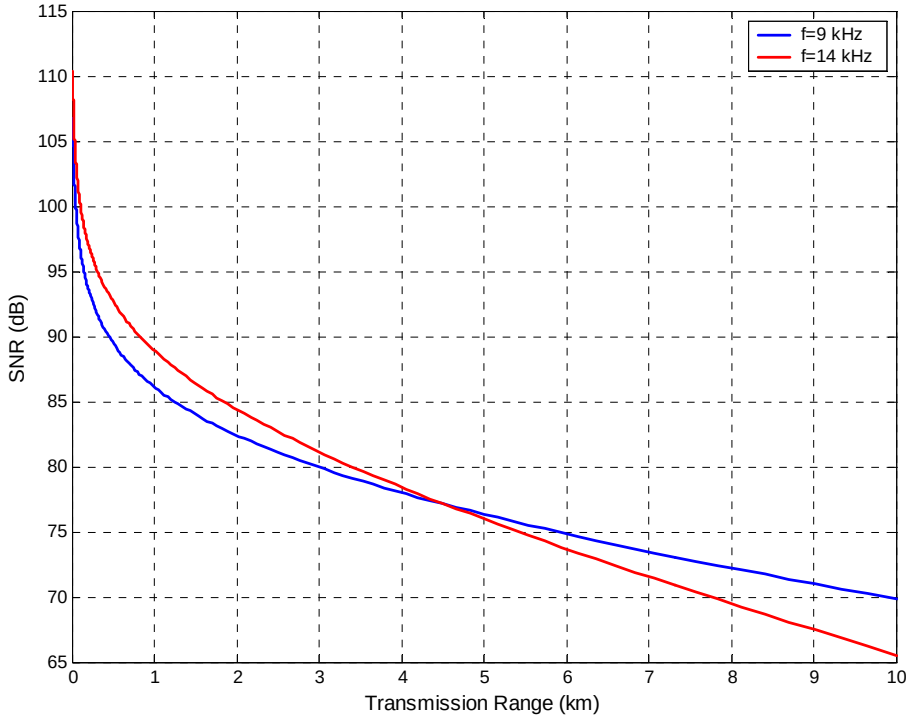

Figure 8. SNR for the Seaweb Operating Frequencies (wind speed 15 m/s, $D = 0.5$).

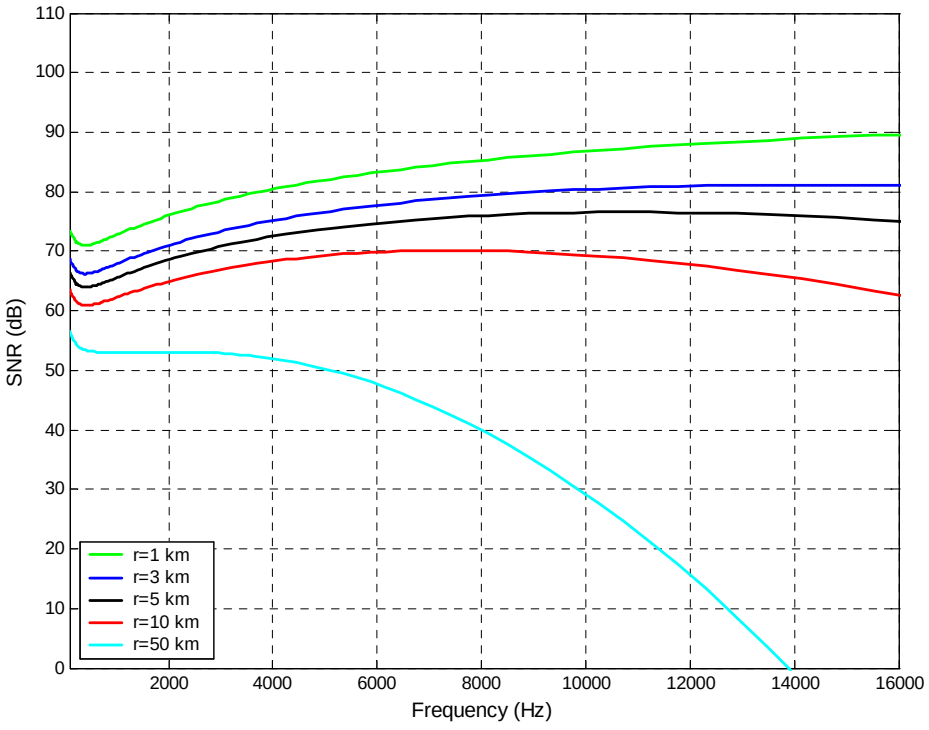

Figure 9. SNR for Different Transmission Ranges (wind speed 15 m/s, $D = 0.5$).

C. TEMPORAL AND SPATIAL VARIABILITY OF THE CHANNEL

The characteristics of the underwater communication channel vary both in time and location. The transmitted signal is subject to multipath propagation and/or Doppler spreading. Together, these factors may cause a severe degradation in the received signal strength.

1. Multipath Propagation

The acoustic wave as it travels through the ocean rarely follows a single path. Due to reflection at the sea boundaries, refraction, and volume scattering within the water column or the sub-bottom structure caused by random inhomogeneities, multiple echoes of the transmitted signal arrive at the receiver. These time-delayed echoes vary both in amplitude and phase and interfere with one another. This effect is called multipath propagation or time spreading since travel time is different for each path. Multipath characteristics are different in shallow and deep-water channels. In shallow water, the multipath is severe because of repeated reflections from the sea surface and bottom. Each reflection causes an attenuation of the sound wave prohibiting long-range communications without a significant loss of acoustic energy [13]. The relative delay time between the first and *i*-th arrival of the transmitted signal at the receiver is called excess delay τ_i. The maximum excess delay $\tau_{\max}$ between the first arrival and the arrival of the last distorted replica of the original signal is much greater in an acoustic communication channel than in a land-based RF channel. This is mainly because sound in water travels slower (≈ 1500 m/s) than the electromagnetic waves travel in the air (3×10^8 m/s). In the frequency domain, the bandwidth over which the channel passes all spectral components with approximately equal gain is called the coherence bandwidth B_C [18] and is a useful tool for comparing different multipath channels:

$$B_C \approx \frac{1}{\tau_{\max}}. \tag{2.20}$$

2. Doppler Spreading

The underwater environment is time variant. The medium itself is constantly moving due to surface and/or internal waves and the source and receiver platform may also move. In shallow water, surface scattering is anticipated to be the most important

contributor to the overall time variability of the channel. Surface scattering is caused by the weather dependent roughness of the sea surface. The frequency of an acoustic wave incident upon the moving surface of the ocean is affected by spectral broadening. This phenomenon is called Doppler spread. The Doppler spread for the case of a sound wave of frequency f reflected at the sea surface is given by [13]

$$B_D = \left(1.75\times10^{-2}\right)\left(\frac{f\, w^{3/2}\cos\theta}{c}\right), \tag{2.21}$$

where B_D is the Doppler spread in Hz, w is the wind speed in m/s, θ is the grazing angle, and c is the speed of sound ($\approx$ 1500 m/s). For a transmission range much greater than the channel depth ($z >> r$), θ is very small and $\cos\theta \approx 1$. In Figure 10, the Doppler spread for a frequency $f = 14$ kHz and wind speed $w = 15$ m/s (sea state 3) is approximately 9 Hz and the frequency spectrum of the reflected wave is $f \pm B_D = 14 \pm 0.009$ kHz.

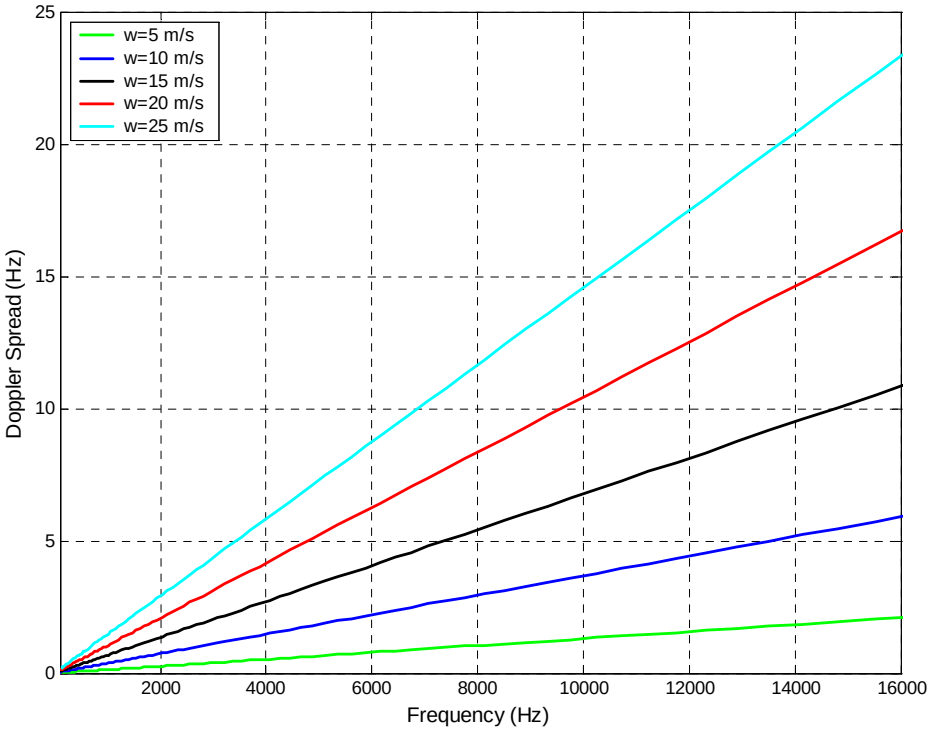

Figure 10. Doppler Spread Due to Surface Reflection.

The second parameter used to describe the time variability of the channel is the coherence time T_C, which is inversely proportional to the Doppler spread (for simplicity the Doppler spread is considered to be the only factor contributing to coherence time),

$$T_C = \frac{1}{B_D}. \tag{2.22}$$

The meaning of coherence time is that two signals arriving at the receiver with a time separation less than T_C will be affected by the channel in the same way [18].

The relation between the aforementioned parameters ($\tau_{\max}$, B_C, B_D, T_C) that describe the overall variability of the channel leads to four distinct types of fading that are summarized in Figure 11:

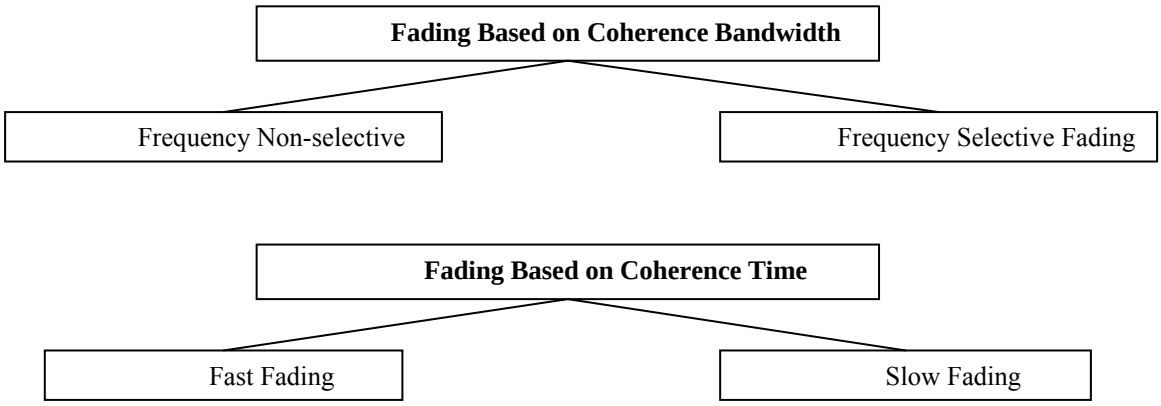

Figure 11. Different Types of Fading Channels [After Ref. 18].

The first two types of fading are frequency selective and frequency non-selective fading. A frequency non-selective channel implies that the coherence bandwidth of the channel B_C is much greater than the bandwidth of the transmitted signal B_S or equivalently the maximum excess delay $\tau_{\max}$ is much less than the digital symbol (bit) period T_b. In this case, which is the best for wireless communications, the channel induces no intersymbol interference (ISI). On the other hand, when frequency selective fading ($B_C < B_S$) occurs, then $T_b < \tau_{\max}$ and ISI is unavoidable.

The other two types of fading are based on the coherence time of the channel. The first one is called a "fast-fading" channel and the second one a "slow-fading" channel. In

a fast-fading communication link the bit duration is greater than the coherence time of the channel. The channel impulse response changes rapidly during the time it takes to transmit one symbol. In this type of fading a severe distortion of the signal occurs and a reliable communication link cannot be maintained. In a slow-fading channel, $T_b << T_C$ ($B_D << B_S$), and less distortion is introduced as a result of the wave propagation in the water.

In this chapter, we have seen the reason the shallow-water channel is considered to be one of the most challenging channels for wireless communications. The system engineer must take into account all the limiting factors (TL, ambient noise, fading) imposed by the channel in order to design a reliable communication link. Because of the time-varying multipath, there is always a trade-off in choosing the transmission rate. A higher bit rate implies greater bandwidth, higher transmission loss and ISI but an almost negligible channel variation over a single-bit interval. In order to undo the effects caused by the channel, a statistical model of the channel impulse response is needed. In the next chapter, we describe the steps needed to extract the channel impulse response by using one of the existing channel models called MMPE, which is based on the parabolic approximation of the wave equation.

III. MODELLING OF THE UNDERWATER CHANNEL

In order to allow for the implementation of an efficient communication scheme, a statistical representation of the underwater channel is needed. In underwater acoustics various numerical methods are used to describe the propagation of sound. All current methods solve the wave equation for a specific environment by following two different theoretical approaches. The first approach is based on wave theory (normal-mode, parabolic equation models) and the second one on ray acoustics (ray models) [12]. Each theory has its own advantages and disadvantages. Wave theory models give a complete solution and are best suited for shallow-water and low-propagating frequencies. Ray models provide a practical visualization of the propagation of sound in space and give their best results at high frequencies [11].

A. MONTEREY-MIAMI PARABOLIC EQUATION (MMPE) MODEL

In this thesis the channel impulse response is predicted using the Monterey-Miami parabolic equation (MMPE) model [7]. The MMPE is a range-dependent, wave- theory model based upon the parabolic approximation of the wave equation. Since the purpose of this research is not the model itself, only an outline of its theoretical foundation is presented. We begin by representing the time harmonic acoustic pressure field in cylindrical coordinates assuming azimuthal symmetry,

$$P(r,z,\omega t) = p(r,z)e^{-i\omega t}. \tag{3.1}$$

Next we substitute Equation (3.1) into the linear, homogeneous wave equation in cylindrical coordinates,

$$\nabla^2 p(r,z) - \frac{1}{c(r,z)^2}\frac{\partial^2}{\partial t^2}p(r,z) = 0, \tag{3.2}$$

where c [m/s] is the speed of sound in water and ∇^2 the Laplacian defined as

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{\partial^2}{\partial z^2}. \tag{3.3}$$

This leads to the time-independent Helmholtz equation,

$$\nabla^2 p(r,z) + \frac{\omega^2}{c(r,z)^2} p(r,z) = 0, \tag{3.4}$$

which can be factored by introducing the operator Q_{op} defined as

$$Q_{op} = \sqrt{\mu + \varepsilon + 1}, \tag{3.5}$$

with

$$\varepsilon = n^2 - 1, \tag{3.6}$$

$$n = \frac{c_0}{c}, \tag{3.7}$$

and

$$\mu = \frac{1}{k_0^2} \frac{\partial^2}{\partial z^2}. \tag{3.8}$$

In the above expressions, n is the index of refraction, c_0 is the reference sound speed of the ocean volume (usually taken as 1500 m/s), and $k_0 = \omega / c_0$ is the reference wavenumber. Note that the coordinate dependency of n and c is dropped for simplicity but is always implied.

If we take into account the cylindrical spreading, which is typical in shallow water and properly factor the Helmholtz equation, the out-going acoustic pressure may then be defined as [19],

$$p(r,z) = P_0 \sqrt{\frac{R_0}{r}} Q_{op}^{-1/2} \psi(r,z) e^{ik_0 r}, \tag{3.9}$$

where P_0 is the pressure amplitude at distance $r = R_0$, and $\psi(r,z)$ is the envelope function, or PE field function. After substituting Equation (3.9) into (3.4) the parabolic equation for the PE field function is formulated:

$$\frac{\partial \psi}{\partial r} = -ik_0 \psi + ik_0 Q_{op} \psi = -ik_0 H_{op} \psi, \tag{3.10}$$

where

$$H_{op} = 1 - Q_{op} \tag{3.11}$$

is an operator, defining the evolution of the PE field function in range. By integrating Equation (3.10) and assuming that H_{op} is constant over a distance Δr, the relationship between the values of ψ at different ranges may be defined as

$$\psi(r + \Delta r) = e^{-i\Delta r k_0 H_{op}(r)} \psi(r), \tag{3.12}$$

where $e^{-i\Delta r k_0 H_{op}(r)}$ is a propagator that accounts for the change in the value of the PE field function ψ with range. This propagator is computed using a split-step Fourier (PE/SSF) method described in [20]. The discrete fast Fourier transform (FFT) subroutine allows the PE/SSF implementation to be represented by [7]

$$\psi(r + \Delta r, z) = e^{-ik_0 \frac{\Delta r}{2} U_{op}(r+\Delta r, z)} \mathrm{FFT}\left\{ e^{-ik_0 \Delta r \hat{T}_{op}(k_z)} \mathrm{IFFT}\left(e^{-ik_0 \frac{\Delta r}{2} U_{op}(r,z)} \psi(r,z) \right) \right\}. \tag{3.13}$$

The U_{op} and $\hat{T}_{op}$ operators are defined as

$$U_{op} = -(n-1), \tag{3.14}$$

and

$$\hat{T}_{op} = 1 - \sqrt{1 - \left(\frac{k_z}{k_0} \right)^2}, \tag{3.15}$$

where k_z is the vertical wavenumber. The output of the model is the PE field function ψ from which other acoustical quantities can be computed. For example by substituting Equation (3.13) into (3.9) we obtain the ocean frequency response,

$$G(f) = p(r,z) = \frac{1}{\sqrt{r}} \psi(r,z) e^{ik_0 r}. \tag{3.16}$$

For broadband signals the MMPE model must be run for each frequency in the band in order to compute the frequency response at a given range and depth. In the following sections of this chapter, the theory behind the various perturbations modeled in this thesis is described and the results of the simulations are presented.

B. INTERFACE ROUGHNESS PERTURBATION

In order to define the interface roughness, we assume a 2-D interface spectrum of the form [21]

$$W_2^{(I)}(k_r) = \frac{\mu}{\left(1 + L_{corr}^2 k_r^2\right)^{\frac{\beta}{2}}}, \tag{3.17}$$

where k_r is the horizontal spatial wavenumber vector given by

$$k_r = \sqrt{K^2 + L^2}, \tag{3.18}$$

K and L are the horizontal wavenumbers in the x- and y-directions, respectively, L_{corr} is a correlation length scale, β is the spectral exponent and μ is a normalization factor defined in terms of the root-mean-square (rms) roughness σ [22],

$$\mu = \frac{1}{\pi}\left(\frac{\beta}{2} - 1\right)\sigma^2 L_{corr}^2. \tag{3.19}$$

For the determination of the interface roughness affecting forward propagation, the one-dimensional (1-D) spectrum along the x-axis $W_1^{(I)}(K)$ is required. This is computed by taking the 1-D transform of the 2-D interface spectrum $W_2^{(I)}(k_r) = W_2^{(I)}(K, L)$ along a slice at $y = 0$, defined by

$$W_1^{(I)}(K) = \int_{-\infty}^{+\infty} W_2^{(I)}(K, L)\, dL. \tag{3.20}$$

In cylindrical coordinates Equation (3.20) yields

$$W_1^{(I)}(K) = \gamma\sigma^2 L_{corr}\left(1 + L_{corr}^2 K^2\right)^{-\frac{\beta}{2}+\frac{1}{2}}, \tag{3.21}$$

where γ is given by

$$\gamma = \frac{\left(\frac{\beta}{2} - 1\right)\Gamma\left(\frac{1}{2}\right)\Gamma\left(\frac{\beta - 1}{2}\right)}{\pi\Gamma\left(\frac{\beta}{2}\right)}, \tag{3.22}$$

and Γ is the Euler gamma function. In order to illustrate the interface roughness perturbation to the underwater environment, the TL vs. depth and range is computed for the frequency f = 11.5 kHz, a bottom depth of 150 m, and a source depth of 75 m. The results are shown in Figure 12 for an rms roughness value of 2 m. By looking at the two plots

two observations can be made. For the case of the unperturbed environment (top plot) the down-refracting sound speed profile causes significant interactions with the seafloor. The upper figure also displays a clear interference pattern due to the coherent summation of modes. On the other hand, when the interface roughness is added, the bottom interactions with the rough boundary create a diffuse field due to the incoherent scatter of energy and the breakdown of coherence in the mode interference pattern.

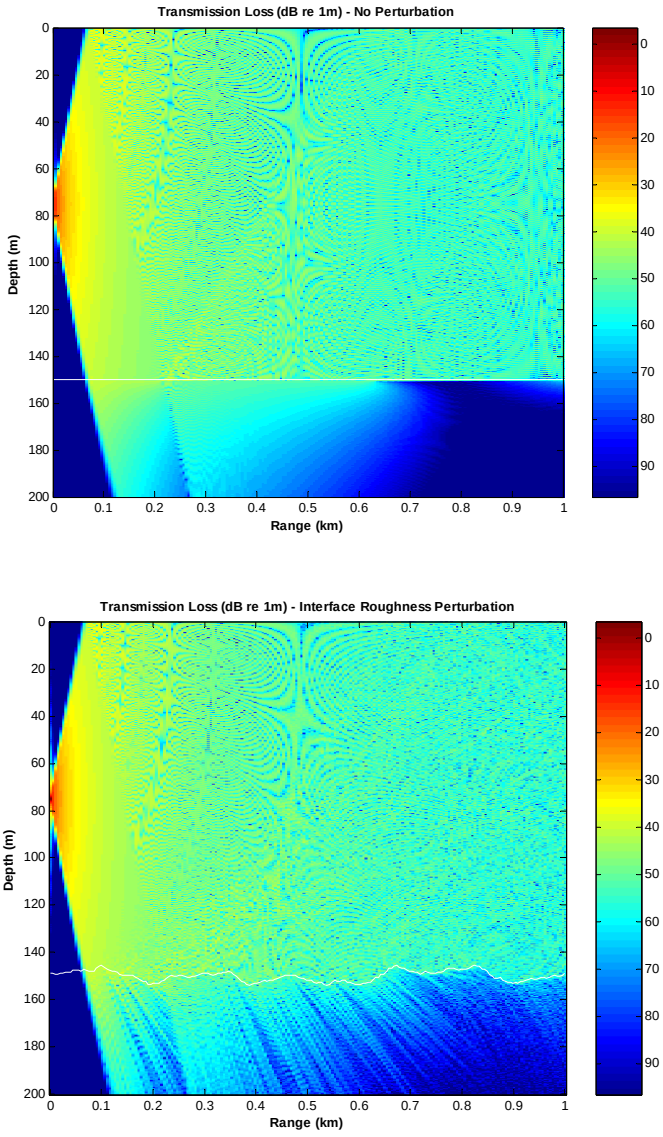

Figure 12. Interface Roughness Perturbation Compared to the Unperturbed Environment (Source Depth = 75 m, f = 11.5 kHz, Range 0 to 1 km).

C. BOTTOM VOLUME PERTURBATION

1. Volume Sound Speed Fluctuations

For the computation of the volume sound speed fluctuations affecting the forward propagation, the two-dimensional (2-D) vertical spectrum of the volume fluctuations $W_{2V}^{(V)}(k_V)$ is required, where $k_V = \sqrt{K^2 + M^2}$ is the wavenumber in the (x,z) plane. This can be determined by first modeling the sediment volume sound speed perturbations with help of the three-dimensional (3-D) volume spectrum $W_3^{(V)}(K,L,M)$ defined by [23]

$$W_3^{(V)}(K,L,M) = \frac{\beta \Lambda^2 B}{2\pi}\left(\Lambda^2\left(K^2 + L^2\right) + M^2\right)^{-\left(\frac{\beta+2}{2}\right)}, \tag{3.23}$$

where M is the vertical wavenumber, B is the spectral strength constant, and Λ is the horizontal-to-vertical aspect ratio used to describe the anisotropy of fluctuations in the sediment.

The 2-D vertical volume spectrum $W_{2V}^{(V)}(K,M)$ in the (x,z) plane is computed by integrating (3.23) over L as follows [22]:

$$W_{2V}^{(V)}(K,M) = \int_{-\infty}^{\infty} W_3^{(V)}(K,L,M)\,dL = \frac{\beta\Lambda^2 B}{\pi}\int_0^{\infty}\left[\Lambda^2\left(K^2 + L^2\right) + M^2\right]^{-\left(\frac{\beta+2}{2}\right)} dL. \tag{3.24}$$

For the specific case of $\beta = 2,\ \Lambda \approx 5,$ and $B \approx 5\times 10^{-4}$ [23], Equation (3.24) yields [22]

$$W_{2V}^{(V)}(K,M) = \int_{-\infty}^{\infty} W_3^{(V)}(K,L,M)\,dL = \alpha'\left[25K^2 + M^2\right]^{-\frac{3}{2}}, \tag{3.25}$$

where $\alpha' = 1.25\times 10^{-3}$.

2. Volume Density Fluctuations

The variability in bottom volume density ρ is treated through the effective index of refraction n' defined by [7]

$$n'^2 = n^2 + \frac{1}{2k_0^2}\left[\frac{1}{\rho}\nabla^2\rho - \frac{3}{2}\left(\frac{1}{\rho}\nabla\rho\right)^2\right], \tag{3.26}$$

where k_0 is the reference wavenumber and n is the index of refraction. By considering the fact that the sediment properties are horizontally stratified, and assuming that the environment is range-independent over a range step Δr, Equation (3.26) simplifies to

$$n'^2 = n^2 + \frac{1}{2k_0^2}\left[\frac{1}{\rho}\frac{\partial^2\rho}{\partial z^2} - \frac{3}{2}\left(\frac{1}{\rho}\frac{\partial\rho}{\partial z}\right)^2\right]. \tag{3.27}$$

The definition of the index of refraction in the bottom is

$$n_b^2 = \frac{c_0^2}{c_b^2}, \tag{3.28}$$

where

$$c_b = c_{b_0}\left(1 + b_z + \delta_l\right) = c_{b_0} + \delta c_b. \tag{3.29}$$

The factor δc_b is given by

$$\delta c_b = c_{b_0}\left(b_z + \delta_l\right), \tag{3.30}$$

where c_{b_0} is the mean bottom sound speed at the interface, δ_l is the zero-mean random perturbation, and $b_z = g/c_{b_0}$ is the normalized gradient of the bottom sound speed.

The relative fluctuations in bottom volume density are related with the relative fluctuations in sound speed through the expression [23]

$$\frac{\delta\rho}{\rho_0} = \frac{2\left(\rho_r - \rho_0\right)}{2\rho_0 - \rho_r}\left(\frac{\delta c}{c_0}\right) = \left(2\gamma\right)\frac{\delta c}{c_0}, \tag{3.31}$$

where

$$c_0 = c_{b_0}\left(1 + b_z\right), \tag{3.32}$$

and

$$\gamma = \frac{\rho_r - \rho_0}{2\rho_0 - \rho_r}. \tag{3.33}$$

In this case, $\rho_r = 2650\,\mathrm{kg/m^3}$ is the density of the grain, and ρ_0, c_0 are the average values of the density and sound speed in the sediment, respectively. From Equation (3.31), we obtain for the bottom-volume density with perturbation,

$$\rho = \rho_0 + \delta\rho = \rho_0 + \rho_0\left(2\gamma\right)\frac{\delta c}{c_0} = \rho_0\left(1 + 2\gamma\frac{\delta c}{c_0}\right). \tag{3.34}$$

The first and the second partial derivatives of (3.34) with respect to depth z (depth gradients in ρ_0 or c_0 are neglected) yield

$$\frac{\partial \rho}{\partial z} = \frac{2\gamma\rho_0}{c_0}\frac{\partial}{\partial z}(\delta c), \tag{3.35}$$

and

$$\frac{\partial^2 \rho}{\partial z^2} = \frac{2\gamma\rho_0}{c_0}\frac{\partial^2}{\partial z^2}(\delta c). \tag{3.36}$$

These are then incorporated into the definition of the effective index of refraction, Equation (3.26), within the bottom volume. The bottom-volume perturbation in terms of the TL vs. depth and range is displayed in Figure 13. As shown in this figure, the scattered acoustic energy at short distances from the source is penetrating into the bottom. However, this has only a minor impact in the resulting interference pattern compared to the one created by the unperturbed environment.

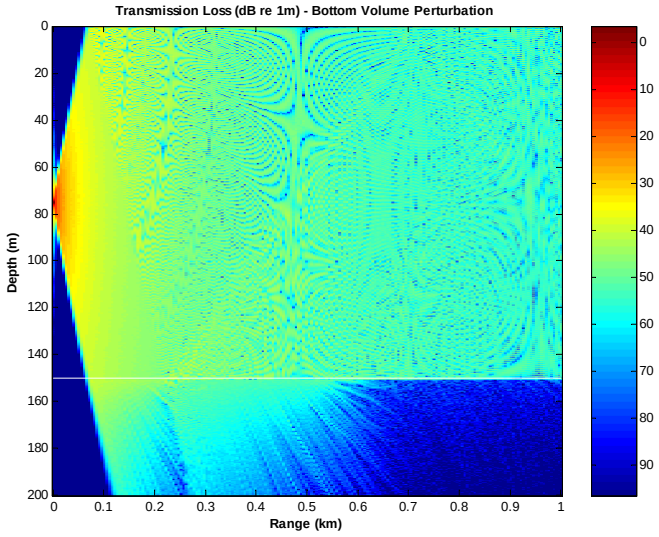

Figure 13. Bottom Volume Perturbation: TL vs. Range and Depth (Source Depth = 75 m, f = 11.5 kHz, Range 0 to 5 km).

D. TURBULENCE PERTURBATION

Another mechanism affecting the coherence structure in an underwater environment is the small-scale fluctuations of sound speed within the water volume caused by turbulence. The turbulent perturbation field is approximated as a random realization of perturbations that are based on an accepted power spectral density of the turbulence.

The variance of the perturbations to the index of refraction $\mu(\vec{r})$ is given by [22]

$$\text{var}(\mu(\vec{r})) = \frac{1}{(2\pi)^3}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S_\mu(K,L,M)\,dK\,dL\,dM = \frac{1}{(2\pi)^3}\int_{-\infty}^{\infty} S_\mu(\vec{k})d^3k, \quad (3.37)$$

where $S_\mu(\vec{k}) = S_\mu(K,L,M)$ is the spectral density of the perturbations and $\vec{k} = (K,L,M)$ is the 3-D wavenumber vector. If the variance is normalized according to

$$\text{var}(\mu(\vec{r})) = \left\langle \left|\mu(\vec{r})\right|^2 \right\rangle = \frac{1}{\text{volume}}\int_V \mu^2(\vec{r})d^3r, \quad (3.38)$$

where $\langle\ \rangle$ denotes averaging , then by combining Equations (3.37) and (3.38) we obtain

$$\int_V \mu^2(\vec{r})d^3r = \frac{\text{volume}}{(2\pi)^3}\int_{-\infty}^{\infty} S_\mu(\vec{k})d^3k. \quad (3.39)$$

Now, if $F_\mu(\vec{k})$ is the perturbation transform, the index of refraction perturbation $\mu(\vec{r})$ can be written as

$$\mu(\vec{r}) = \frac{1}{(2\pi)^3}\int_{-\infty}^{\infty} F_\mu(\vec{k})e^{i\vec{k}\cdot\vec{r}}d^3k, \quad (3.40)$$

and its spatial autocorrelation is given by

$$\int_{-\infty}^{\infty}\mu(\vec{r})\mu^*(\vec{r}-\vec{r}')d^3r = \frac{1}{(2\pi)^6}\int_{-\infty}^{\infty}\left\{\left[\int_{-\infty}^{\infty} F_\mu(\vec{k})e^{i\vec{k}\cdot\vec{r}}d^3k\right]\left[\int_{-\infty}^{\infty} F_\mu^*(\vec{k}')e^{-i\vec{k}\cdot(\vec{r}-\vec{r}')}d^3k\right]\right\}d^3r. \quad (3.41)$$

For $\vec{r}' = 0$, Equations (3.39) and (3.41) yield the desired relation between the signal spectrum $F_\mu(\vec{k})$ and the 3-D spectral density of the perturbations $S_\mu(\vec{k})$ [22]:

$$\left|F_\mu(\vec{k})\right| = \left[\text{volume}\times S_\mu(\vec{k})\right]^{1/2}. \quad (3.42)$$

Similarly, along the 2-D plane of propagation $(\vec{r} = (x,0,z))$, the relation between the 2-D spectral density $V_\mu(K,M)$, and the 2-D spectrum of the perturbation field $G_\mu(K,M)$, can be written as [22]

$$\left|G_\mu(K,M)\right| = \left[\text{Area}\times V_\mu(K,M)\right]^{1/2}. \quad (3.43)$$

The turbulent perturbation field is approximated through the use of the method of structure functions [24], in which large-scale perturbations are not accounted for. The three-dimensional spectral density function $E(k)$, can be expressed as [25]

$$E(k) = A\varepsilon^{2/3}k^{-11/3}, \tag{3.44}$$

where $k^{-11/3} = \left(K^2 + L^2 + M^2\right)^{-11/6}$, ε is the energy dissipation rate, and A is a scalar multiplier. In order to prevent $E(k)$ from growing unbounded when the value of k is small, a lower wavenumber threshold k_t is defined so that the $k^{-11/3}$ dependence is [22]

$$E(k) \approx \frac{1}{(k^2 + k_t^2)^{11/6}}. \tag{3.45}$$

Also, by applying the high-pass filter

$$\text{HPF} = \frac{k^2}{k^2 + k_t^2}, \tag{3.46}$$

the 3-D spectral density function $E(k)$ is forced to assume the value of zero when $k = 0$. Subsequently, a high-frequency cutoff is established expressed as [25]

$$R_B(k) = \exp\left[-q\left(\frac{k}{k_B}\right)^2\right], \tag{3.47}$$

where $R_B(k)$ is the Batchelor spectrum, q is an order unity factor, and k_B is the Batchelor wavenumber defined by [22]

$$k_B = \left(\frac{\overline{\varepsilon}}{\nu\,\kappa_T^2}\right)^{1/4}. \tag{3.48}$$

In the above expression, $\overline{\varepsilon}$ $\left[\text{W/kg}\right]$ is the depth averaged kinetic energy dissipation rate, $\kappa_T \approx 10^{-7}\ \text{m}^2/\text{s}$ is the thermal diffusivity of the sea water, and $\nu = 1.4\times10^{-6}\ \text{m}^2/\text{s}$ is the kinematic viscosity. In this theoretical development, $\overline{\varepsilon}$ is given by [26]

$$\overline{\varepsilon} = \frac{0.3}{f_I}\left\langle\left[\frac{4}{\pi}(j_*)N(z)(b\text{E}_{GM})f_I\right]^2 \cosh^{-1}\left(\frac{N(z)}{f_I}\right)\right\rangle_Z, \tag{3.49}$$

where f_I $\left[\text{rad/s}\right]$ is the inertial frequency, $b\text{E}_{GM} \approx 0.5\,\text{m}$ is a measure of internal wave intensity, j_* is the mode number, and $N(z)$ $\left[\text{rad/s}\right]$ is the buoyancy frequency expressed in terms of density $\rho(z)$ and potential density $\rho_p(z)$,

$$N(z) = g^{1/2}\left[\frac{1}{\rho(z)}\frac{\partial\rho_p(z)}{\partial z}\right]^{1/2}. \tag{3.50}$$

By following the theoretical approach described above, the three-dimensional spectral density $S_\mu(k;z)$ can now be expressed by [22]

$$S_\mu(k;z) = \frac{A_\mu}{(k^2 + k_t^2)^{11/6}} \left(\frac{k^2}{k^2 + k_t^2} \right) R_B(k), \tag{3.51}$$

where A_μ [$m^{-2/3}$] is the turbulent strength parameter. In this implementation, A_μ will be adjusted manually to produce a specific desired rms fluctuation.

The desired 2-D spectral density is now formulated by integrating over L ($y = 0$) only the portion of the 3-D spectral density that varies with the turbulent spectrum. This portion is given by [22]

$$\Phi_{3D}(k) = \frac{1}{(k^2 + k_t^2)^{11/6}}. \tag{3.52}$$

Thus, the 2-D spectral density $\Phi_{2D}(k)$ can be derived from

$$\Phi_{2D}(K,M) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{1}{(k^2 + k_t^2)^{11/6}} dL = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{1}{(K^2 + L^2 + M^2 + k_t^2)^{11/6}} dL$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{1}{(\alpha^2 + L^2)^{11/6}} dL = \frac{1}{\pi \alpha^{11/3}} \int_0^{\infty} \frac{1}{\left(1 + \frac{L^2}{\alpha^2}\right)^{11/6}} dL, \tag{3.53}$$

where $\alpha^2 = K^2 + M^2 + k_t^2$. With help of integral tables, Equation (3.53) yields

$$\Phi_{2D}(K,M) = \left(\frac{1}{2\pi \alpha^{8/3}} \right) \frac{\sqrt{\pi}\,\Gamma(\frac{4}{3})}{\Gamma(\frac{11}{6})}. \tag{3.54}$$

By reapplying the terms that do not vary with the turbulent spectrum, the resulting 2-D spectral density $\Phi_{2D}(k)$ is defined by [22]

$$V_\mu(k;z) = \left(\frac{\sqrt{\pi}}{2\pi} \right) \left(\frac{\Gamma\left(\frac{4}{3}\right)}{\Gamma\left(\frac{11}{6}\right)} \right) \frac{A_\mu}{(k^2 + k_t^2)^{4/3}} \left(\frac{k^2}{k^2 + k_t^2} \right) R_B(k). \tag{3.55}$$

The perturbation to the underwater environment because of turbulence is illustrated in Figure 14 for an rms value of 2 m/s. The breakdown in the coherent structure of the modes is obvious, resulting in a highly diffuse field with random patterns of ray-like propagation (yellow, high-energy paths in Figure 14).

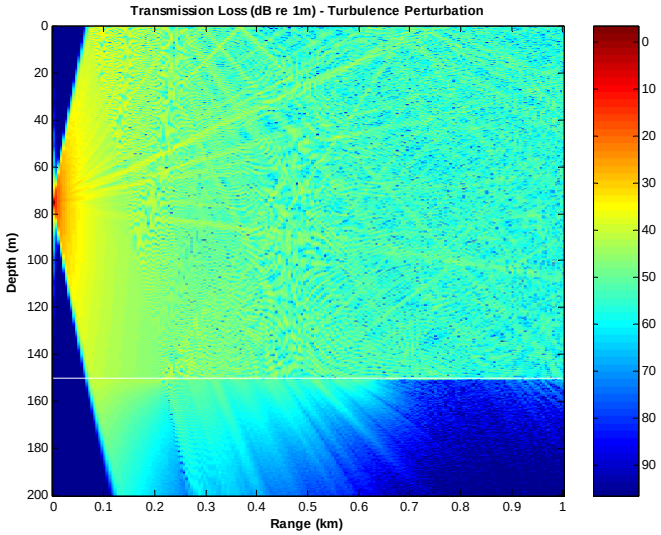

Figure 14. Turbulence Perturbation: TL vs. Range and Depth (Source Depth = 75 m, f = 11.5 kHz, Range 0 to 5 km).

E. INTERNAL WAVE PERTURBATION

Larger-scale fluctuations in sound speed may also be generated from internal waves. The internal wave perturbation accounts for the vertical movement of water masses inside the sea body due to buoyancy forces. In this thesis a combination of a linear (sinusoidal) and a nonlinear soliton internal wave perturbation to the sound speed is modeled. The linear internal wave perturbation is expressed as a sum of sinusoids [27],

$$\Delta c_{\sin}(r,z) = C\frac{z}{B}\exp\left(-\frac{z}{B}\right)\sum_{i=1}^{5}\cos(K_i r), \tag{3.56}$$

where $\Delta c_{\sin}(r,z)$ is the sound speed perturbation at a given range r and depth z, $B = 25$ m, $K_i = 2\pi/\lambda_i$, $\lambda_i = 1500 - 300(i-1)$ m, and C is defined in such a way that the maximum value of $\Delta c_{\sin}(r,z)$ is 1.5 m/s.

The nonlinear soliton internal wave perturbation $\Delta c_{\text{sol}}(r,z)$ is defined as [27]

$$\Delta c_{\text{sol}}(r,z) = C\frac{z}{B}\exp\left(-\frac{z}{B}\right)\sum_{i=1}^{6}A_i\left[\operatorname{sech}\left(\frac{R_i - r}{D_i}\right)\right]^2, \tag{3.57}$$

where $B = 25$ m, $A_i = 10\exp(-0.3(i-1))$, $R_1 = 4800$ m, $R_{i>1} = R_{i-1} - 200(7-i)$ m, $D_i = \sqrt{34300/A_i}$ m, and C is defined such that the maximum value of the sound speed

perturbation is 12.5 m/s. The internal wave perturbation in terms of the TL vs. depth and range is shown in Figure 15 (range 0 to 1 km) and Figure 16 (range 4 to 5 km). At short ranges the perturbation is mostly due to the linear internal waves and the modes seem to remain coherent. At long ranges from the source (Figure 16) the nonlinear soliton internal waves cause a much stronger perturbation to the field resulting in a more random like dispersion of the energy.

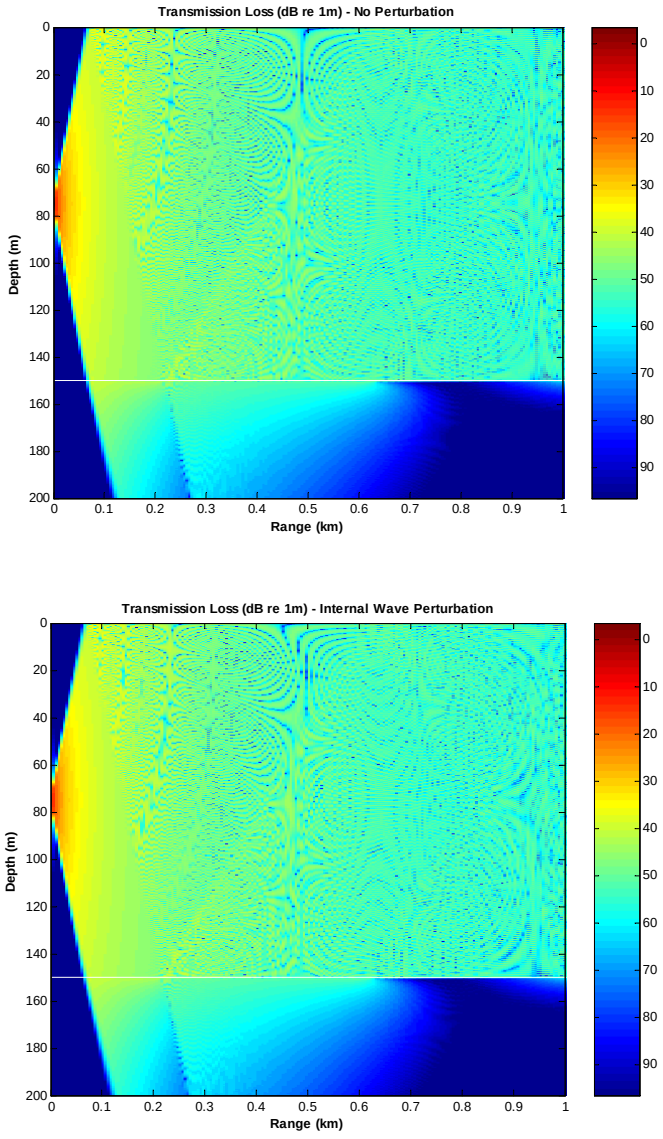

Figure 15. Internal Wave Perturbation Compared to the Unperturbed Environment (Source Depth = 75 m, f = 11.5 kHz, Range 0 to 1 km).

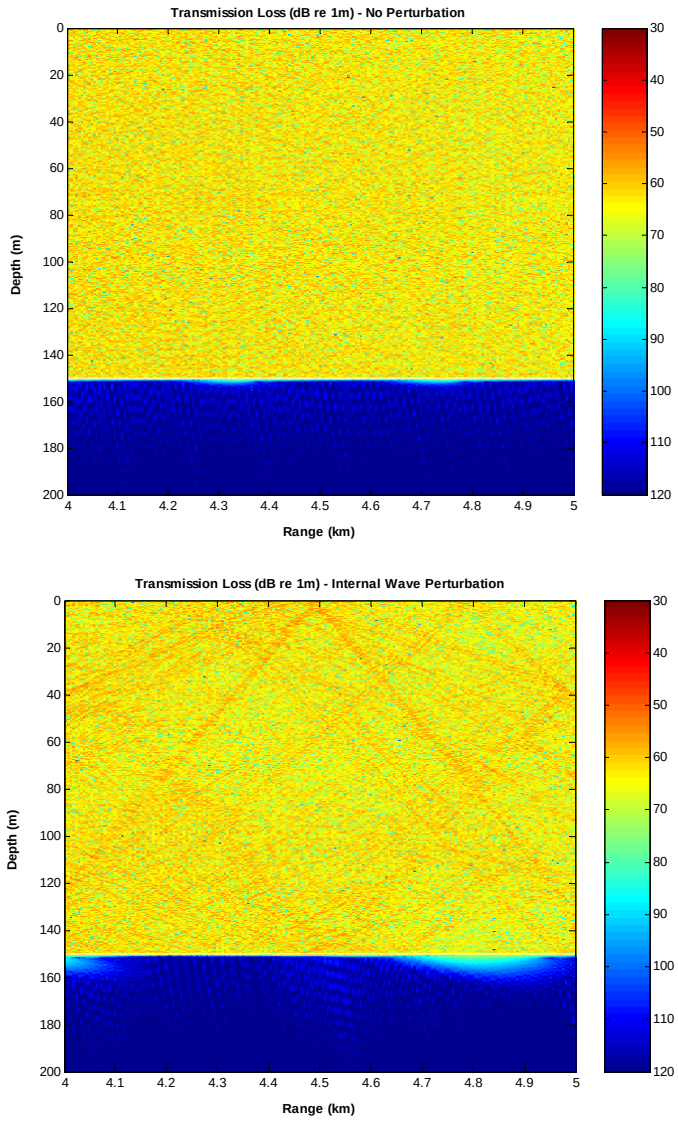

Figure 16. Internal Wave Perturbation Compared to the Unperturbed Environment (Source Depth = 75 m, f = 11.5 kHz, Range 4 to 5 km).

F. DOPPLER PERTURBATION

The influence of Doppler due to source/receiver motion is of great interest in underwater acoustic communications. Because of the Doppler effect the frequency of a transmitted signal suffers from spectral broadening. The actual frequency $f_s(\theta)$, propagated from a source that is moving with speed υ_S, can be expressed as [28]

$$f_s(\theta) = \frac{f_T}{1-\left(\frac{\upsilon_S}{c}\right)\cos(\theta-\phi_S)} \overset{\upsilon_S/c \ll 1}{\approx} f_T\left[1+\left(\frac{\upsilon_S}{c}\right)\cos(\theta-\phi_S)\right], \tag{3.58}$$

where f_T is the transmitted frequency, c is the sound speed, θ is the angle of propagation (relative to the horizontal), and ϕ_S is the angle of source motion. As the sound travels through the medium it follows multiple paths. Therefore, even a single transmitted frequency will result in multiple received frequencies at the receiver.

If the bandwidth BW_T of the transmitted signal is

$$BW_T = f_{T,\max} - f_{T,\min}, \tag{3.59}$$

where $f_{T,\min}$ and $f_{T,\max}$ are the minimum and maximum frequency components of the transmission, then the Doppler shifted bandwidth for forward propagation is given by [28]

$$BW_S^{r+} = BW_T + \frac{\upsilon_S}{c}\left(f_{T,\max} + f_{T,\min}\left|\sin\left(\phi_S\right)\right|\right), \tag{3.60}$$

or

$$BW_S^{r-} = BW_T + \frac{\upsilon_S}{c}\left(f_{T,\max}\left|\sin\left(\phi_S\right)\right| + f_{T,\min}\right). \tag{3.61}$$

Equation (3.60) accounts for a source motion toward the receiver ($r+$) whereas (3.61) is for a source motion away from the receiver ($r-$).

If the receiver is moving also, a similar theoretical approach yields [28]

$$BW_R^{r+} = BW_P + \frac{\upsilon_R}{c}\left(f_{P,\max} + f_{P,\min}\left|\sin\left(\phi_R\right)\right|\right), \tag{3.62}$$

or for the case when the receiver is moving toward the source

$$BW_R^{r-} = BW_P + \frac{\upsilon_R}{c}\left(f_{P,\max}\left|\sin\left(\phi_R\right)\right| + f_{P,\min}\right), \tag{3.63}$$

where υ_R is the receiver speed, ϕ_R is the angle of receiver motion, $f_{P,\min}$ and $f_{P,\max}$ are the minimum and maximum frequency of propagation, and BW_P is the bandwidth of propagation given by

$$BW_P = f_{P,\max} - f_{P,\min}. \tag{3.64}$$

G. RESULTS

The results of the various perturbations described above are presented in this section. The specific parameters defining the characteristics of the underwater environment, the source/receiver, and the resolution of the calculations are summarized in Table 1. For each simulation the MMPE model was run over a 2048-Hz bandwidth (1024 frequencies), and the channel impulse response was calculated for a source-receiver distance of 5 km and a source depth of 75 m as in a typical Seaweb setup. For all modeled perturbations, the transmission loss is presented as a function of depth and frequency (Figure 17 through Figure 23 - top plot). The magnitude of the channel impulse response at a depth of 75 m versus the travel time of the acoustic wave is illustrated in Figure 17 through Figure 23 (bottom plot). The impulse responses are normalized by the magnitude of the unperturbed response such that its values lie between -1 and 1.

From the results presented in the following figures some interesting observations can be made. The typical excess delay $\tau_{\max}$ between the first and last arrival of the transmitted signal is about 0.15 seconds (150 ms). For a symbol duration $T_s = 0.42\,\text{ms}$ corresponding to a chipping rate of 2400 chips-per-second (used in the DS-SS implementation described in Chapter V) channel-induced ISI will occur ($T_s < \tau_{\max}$).

Among the individual perturbations examined, the one corresponding to the interface roughness (Figure 18) seems to be generating the worst fading environment. This was expected since, as already mentioned, in shallow-water channels the interface roughness is typically the dominant scattering factor.

Figure 23 represents the worst-case scenario where all perturbations are combined and Doppler $\left(\upsilon_S = 15\,\text{knots} \approx 8\,\text{m/s}\right)$ is added to account for the spectral broadening induced by the channel. The effect of Doppler is obvious by comparing the results illustrated in Figure 22 (all perturbations and no Doppler) and Figure 23. It is obvious that the underwater channel is far more challenging for communications when Doppler is included.

Parameter	Value	Remarks
Main Parameters		
Number of depth points saved	3277	
Minimum depth saved	0 m	
Maximum depth saved	200 m	
Number of range steps saved	10	
Minimum range saved	0 km	
Maximum range saved	5 km	
Computational range step size	0.4 m	
Maximum computational depth	500 m	
Number of computational depth points	16,384	Radix-2 integer required for FFT
Reference sound speed	1500 m/s	
Source Data		
Source depth	75 m	
Center frequency	11.5 kHz	
Frequency bandwidth	2048 Hz	
Number of Frequencies	1024	Radix-2 integer required for FFT
Sound Speed Profile Data		
Water column sound speed (unperturbed)	1500 m/s	Isospeed
Bathymetry of the Water-Bottom Interface		
Bottom depth (unperturbed)	150 m	Flat
Acoustic Parameters of the Medium below the Water-Bottom Interface		
Bottom sound speed	1600 m/s	
Sound speed gradient	1/s	
Relative density (bottom/water)	1.6	
Compressional attenuation	0.1 dB/km/Hz	
Shear speed	0 m/s	
Shear attenuation	0 dB/m/kHz	
Acoustic Properties of the Deep Layer		
Sub-bottom sound speed	2000 m/s	
Sound speed gradient	1/s	
Relative density (sub-bottom/water)	3 g/cm^3	
Compressional attenuation	0.25 dB/km/Hz	
Shear speed	0 m/s	
Shear attenuation	0 dB/m/kHz	

Parameter	Value	Remarks
Perturbation Parameters		
Source speed	15 knots (8 m/s)	Doppler perturbation
Bottom interface roughness (rms)	2 m	Bottom interface perturbation
Bottom volume sound speed variability (rms)	15 m/s	Bottom volume perturbation (with corresponding density perturbation, as described in text)
Internal wave (linear) sound speed variability (length scales & max amplitudes)	1500 m & 1.5 m/s 1200 m & 1.5 m/s 900 m & 1.5 m/s 600 m & 1.5 m/s 300 m & 1.5 m/s	First part of IW perturbation (always combined with second part of IW)
Internal wave (soliton) sound speed variability (ranges & max amplitudes)	4800 m & 12.5 m/s 3800 m & 9.3 m/s 3000 m & 6.9 m/s 2400 m & 5.1 m/s 2000 m & 3.8 m/s 1800 m & 2.8 m/s	Second part of IW perturbation (always combined with first part of IW)
Water column turbulence (length scale and rms) sound speed variability	100 m & 2 m/s	Turbulence perturbation

Table 1. MMPE Parameters.

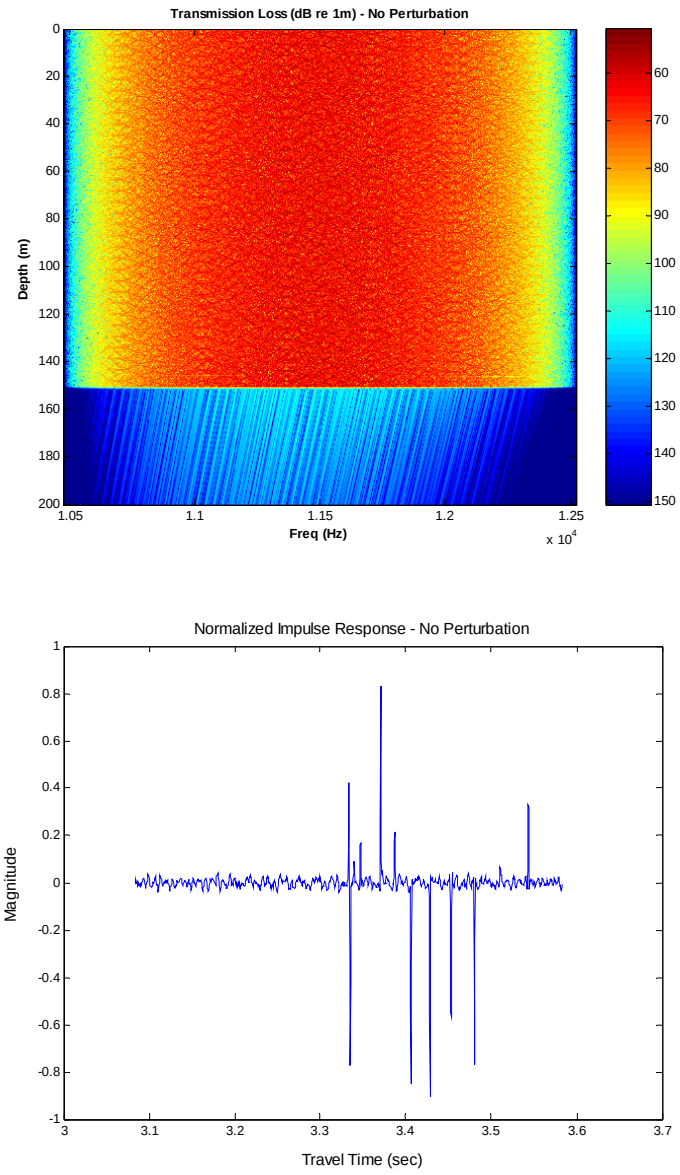

Figure 17. No Perturbation: TL as a Function of Depth and Frequency (top plot) and Normalized Impulse Response at Depth 75 m vs. Travel Time (bottom plot).

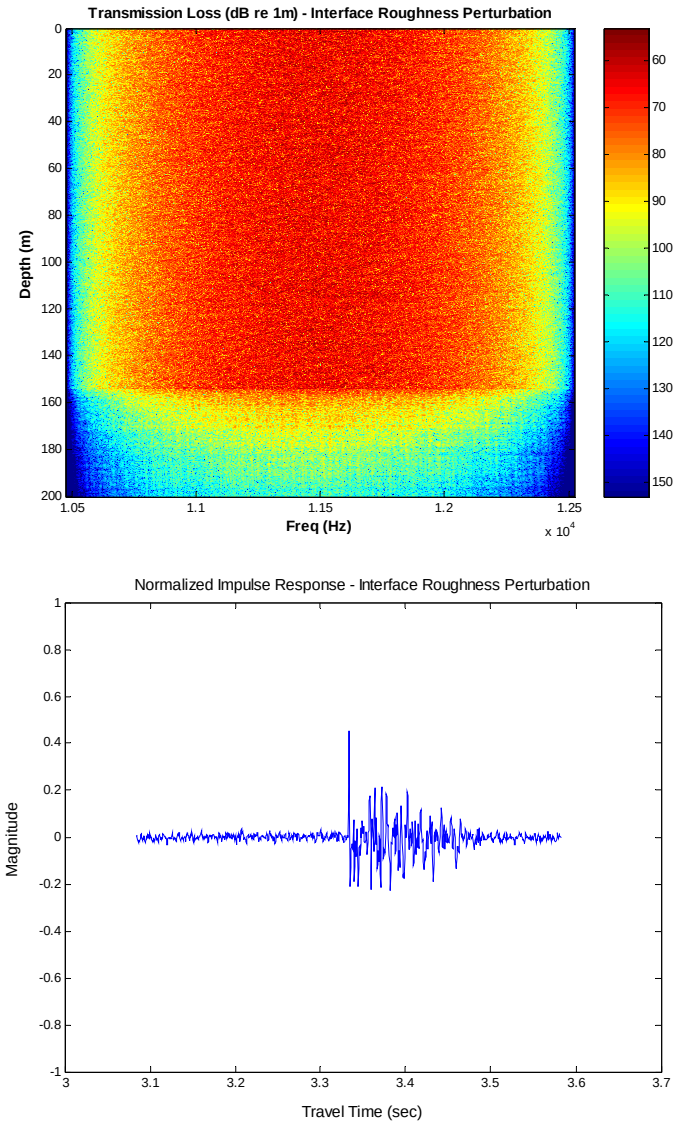

Figure 18. Interface Roughness Perturbation: TL as a Function of Depth and Frequency (top plot) and Normalized Impulse Response at Depth 75 m vs. Travel Time (bottom plot).

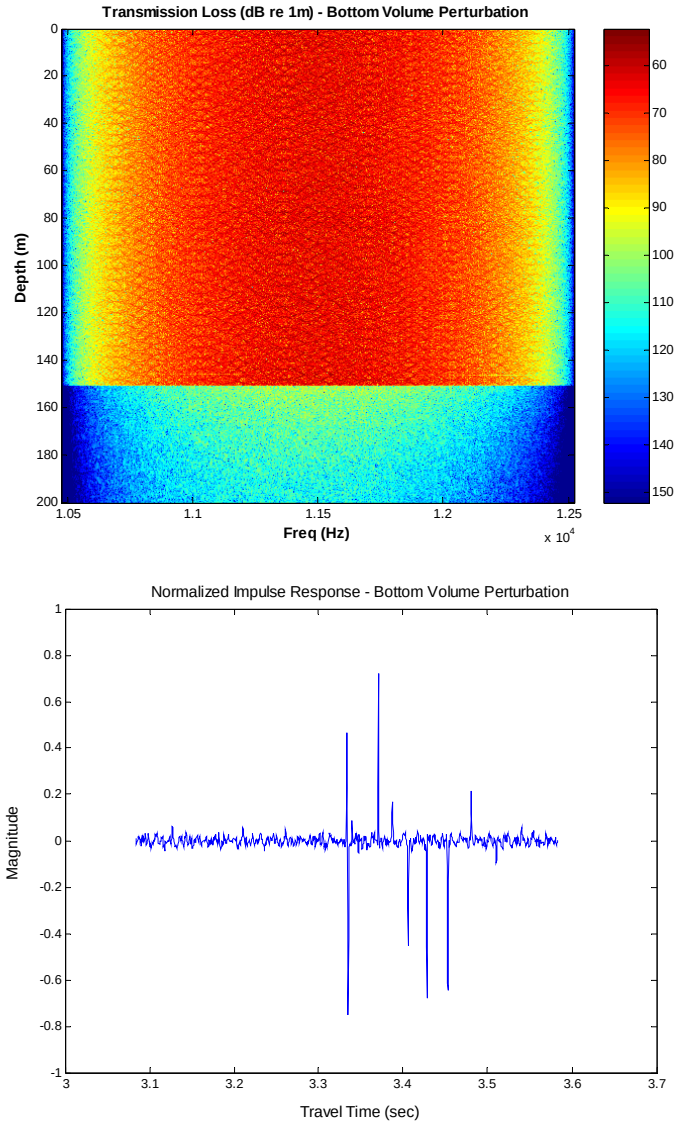

Figure 19. Bottom Volume Perturbation: TL as a Function of Depth and Frequency (top plot) and Normalized Impulse Response at Depth 75 m vs. Travel Time (bottom plot).

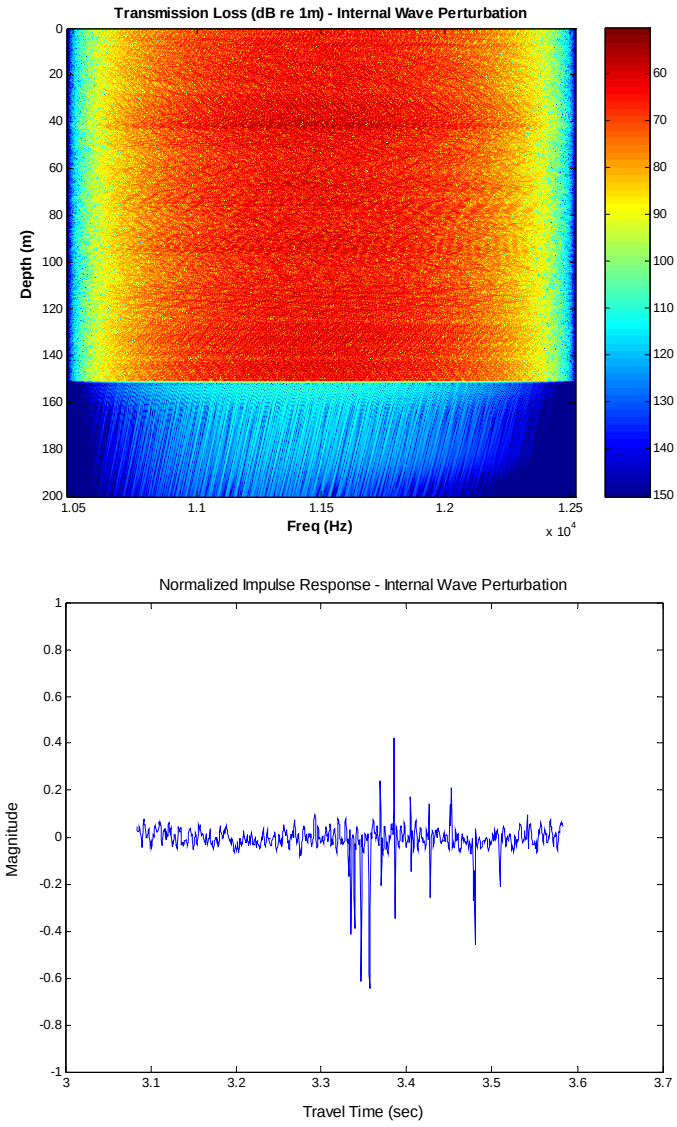

Figure 20. Internal Wave Perturbation: TL as a Function of Depth and Frequency (top plot) and Normalized Impulse Response at Depth 75 m vs. Travel Time (bottom plot).

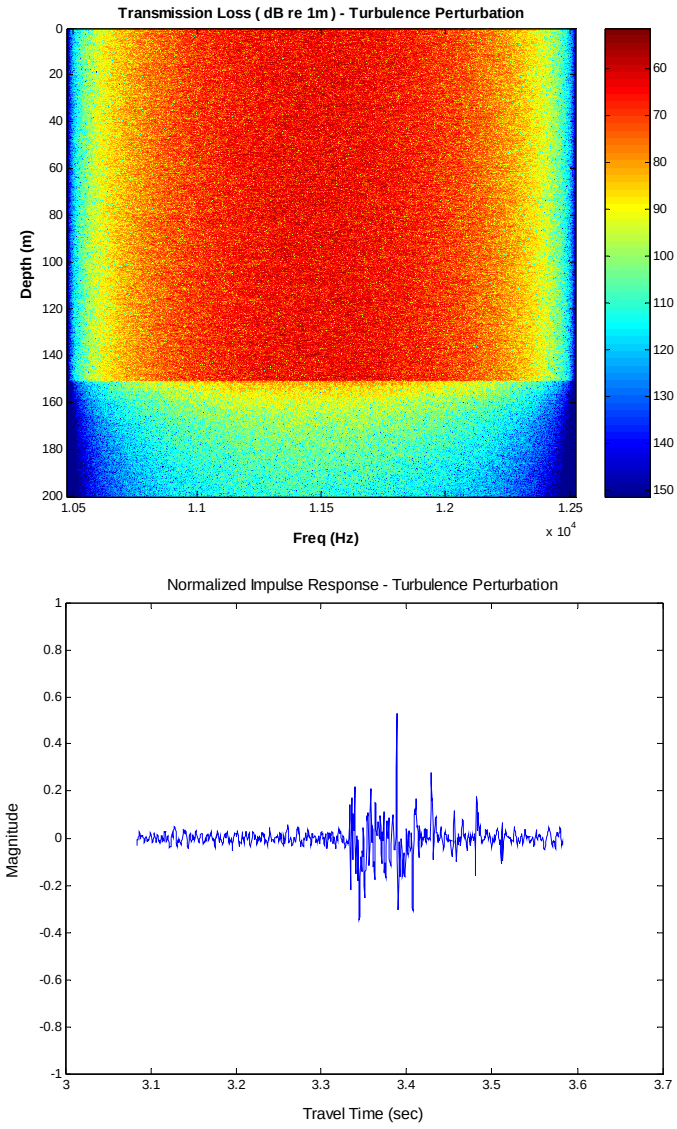

Figure 21. Turbulence Perturbation: TL as a Function of Depth and Frequency (top plot) and Normalized Impulse Response at Depth 75 m vs. Travel Time (bottom plot).

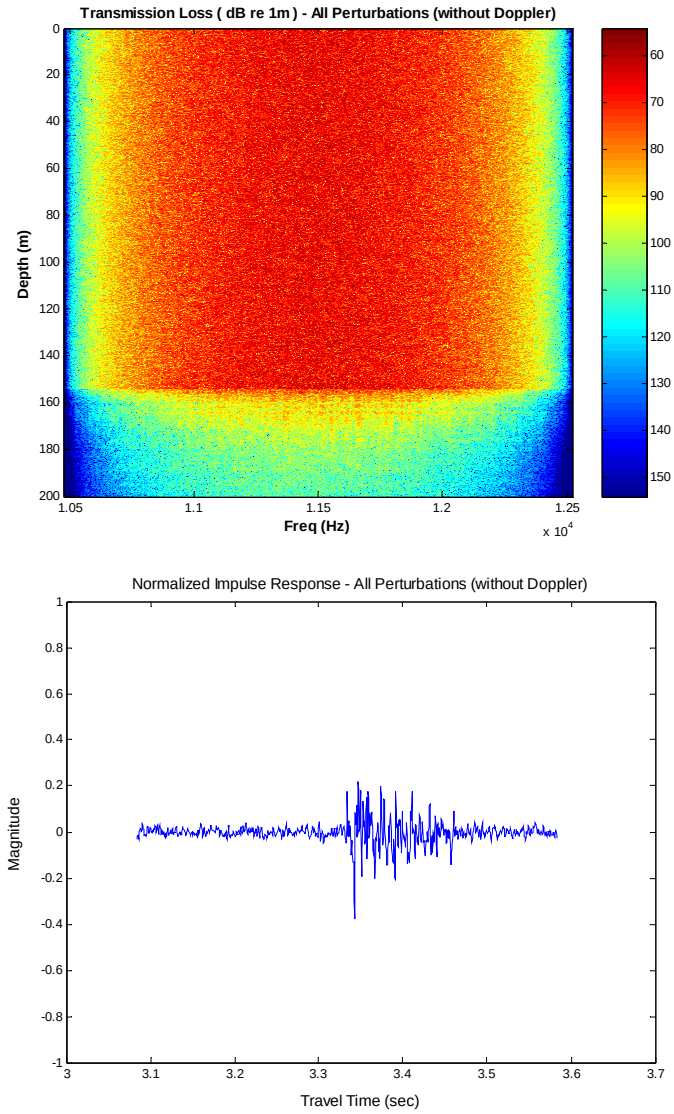

Figure 22. All Perturbations Combined (no Doppler): TL as a Function of Depth and Frequency (top plot) and Normalized Impulse Response at Depth 75 m vs. Travel Time (bottom plot).

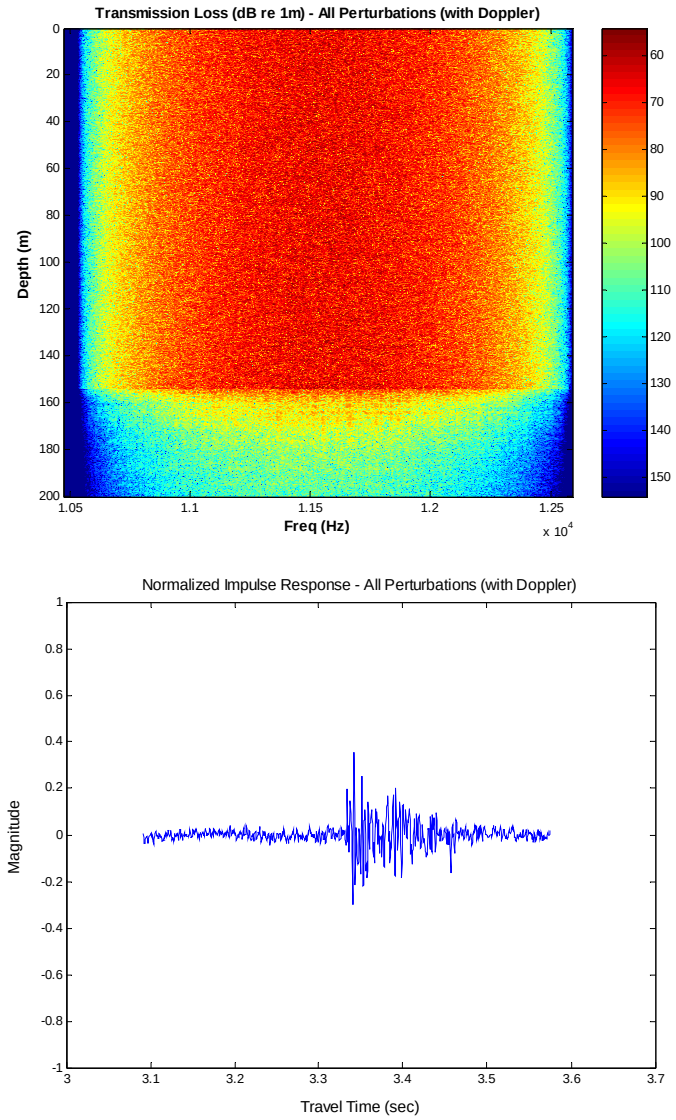

Figure 23. All Perturbations Combined (with Doppler): TL as a Function of Depth and Frequency (top plot), and Normalized Impulse Response at Depth 75 m vs Travel Time (bottom plot).

This chapter presented a brief description of the MMPE model and an outline of the theory behind the various perturbations modeled. The MMPE model was used to predict the channel impulse response for each individual perturbation so that the implementation of the DS-SS scheme with the CRV updating algorithm could be evaluated in Chapter V. In the following chapter the CRV decomposition and its respective updating algorithm is presented, and the reason that the CRV can be used as an alternative to the eigenvalue decomposition (EVD) at less computational cost is shown.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. THE CRV DECOMPOSITION

The purpose of this chapter is to describe the "Cross-product RV decomposition" [9]. The cross-product RV or CRV is a relatively new subspace tracking algorithm based on the URV decomposition [29]. In various signal processing applications, but mostly in noise suppression techniques, it is often needed to separate the signal subspace from the noise subspace. The most powerful tool to perform this task, as far as computational accuracy is concerned, is the singular value decomposition (SVD). Although the SVD is a very reliable means of determining the numerical rank of a correlation matrix, it has an important drawback. SVD is computationally expensive, especially for large matrices, and it is nonrecursive. This significant disadvantage can be effectively addressed by the CRV.

A. CRV DECOMPOSITION

We start by considering an $N \times M$ $(N \geq M)$ data matrix X of numerical rank r. Here, numerical rank means the number of singular values that are larger than a certain threshold. Therefore if $\sigma_1, \sigma_2, ..., \sigma_M$ are the singular values of X, then

$$\sigma_1 \geq \sigma_2 \geq ..... \geq \sigma_r \geq \text{threshold} >> \sigma_{r+1} \approx \sigma_{r+2} \approx ..... \approx \sigma_M \approx 0. \tag{4.1}$$

By definition the singular values σ_i are directly related to the eigenvalues λ_i of the data correlation matrix $X^H X$ since $\lambda_i = \sigma_i^2$. Thus the number of eigenvalues above the specified threshold is the numerical rank of the matrix. In signal processing if the gap at the threshold is well determined, we say that the r largest eigenvalues are associated to the signal subspace and the $M - r$ smaller eigenvalues are associated to the noise subspace. In this way, each subspace is spanned by the corresponding eigenvectors. The CRV decomposition as already mentioned is based upon the URV algorithm [29], which is a rank-revealing decomposition proposed as an alternative to the SVD. The URV decomposition can provide a reliable estimate of the desired subspaces at a reduced computational cost and may be written as

$$X = URV^H = U \begin{bmatrix} E & F \\ 0 & G \end{bmatrix} V^H, \tag{4.2}$$

where the matrices U, V are unitary, the matrices R, E, G are upper triangular, and F is a matrix of very small numbers. The smallest singular value (or eigenvalue) of the E matrix is σ_r and the signal subspace is concentrated in this matrix. The noise subspace is concentrated mainly in the G matrix.

The CRV decomposition is formulated by taking the product of the URV and its transpose. If as before X is an $N \times M$ data matrix, then [9]

$$\Phi = X^H X = \left(URV^H\right)^H \left(URV^H\right) = VR^H \left(U^H U\right) RV^H. \tag{4.3}$$

Considering that U is a unitary matrix $\left(UU^H = I\right)$ and by setting

$$W = R^H R, \tag{4.4}$$

Equation (4.3) can be written as

$$\Phi = VWV^H. \tag{4.5}$$

Equation (4.5) is a cross-product RV decomposition. By combining (4.2) and (4.4) the W matrix is written as

$$W = R^H R = \begin{bmatrix} E & F \\ 0 & G \end{bmatrix}^H \begin{bmatrix} E & F \\ 0 & G \end{bmatrix} = \begin{bmatrix} E^H & 0 \\ F^H & G^H \end{bmatrix} \begin{bmatrix} E & F \\ 0 & G \end{bmatrix} = \begin{bmatrix} E^H E & E^H F \\ F^H E & F^H F + G^H G \end{bmatrix}. \tag{4.6}$$

Finally if $A = E^H E,\ B = E^H F$ and $C = F^H F + G^H G$ the rank-revealing CRV decomposition is formulated

$$\Phi = VWV^H = V \begin{bmatrix} A & B \\ B^H & C \end{bmatrix} V^H. \tag{4.7}$$

Since the smallest singular value of E is σ_r, the smallest eigenvalue of $A = E^H E$ is $\lambda_r = \sigma_r^2$. Therefore the signal subspace power will be concentrated mainly in matrix A and the noise subspace in matrix C.

B. CRV UPDATING ALGORITHM

In many signal processing applications, the rows of the data matrix X correspond to samples of a received signal corrupted with noise and the columns to different signals in time. If the signal and noise subspaces have already been estimated and a new signal is

received, then it would be more convenient to use the previously estimated noise power to approximate the new subspaces than to compute everything from the beginning. This procedure is called updating [29] and can be implemented recursively. The CRV updating algorithm described next is a simplified version of the algorithm proposed in [9].

We start our description by assuming that the decomposition $\Phi = VWV^H$ is known and the updated one $\tilde{\Phi}$ needs to be computed. The updated decomposition is given by

$$\tilde{\Phi} = \tilde{V}\tilde{W}\tilde{V}^H = \alpha\Phi + xx^H, \tag{4.8}$$

where α is the fading (or forgetting) factor with values between 0 and 1 and x the newly received $M \times 1$ data vector. In our implementation, the fading factor is taken to be 1. Thus it can be omitted for the rest of the algorithm description. The next step is to define the intermediate vector z given by

$$z = V^H x, \tag{4.9}$$

so that equation (4.8) may be written as,

$$\tilde{\Phi} = VWV^H + Vz\left(Vz\right)^H = VWV^H + Vzz^HV^H = V\left(W + zz^H\right)V^H = V\Psi V^H, \tag{4.10}$$

and the problem is transformed into updating the matrix $\Psi = W + zz^H$. If the numerical rank of W according to a user specified threshold is r then Ψ is of the form (example for $r = 3$):

$$\Psi = \underbrace{\begin{bmatrix} x & x & x & e & e & \cdots & e \\ x & x & x & e & e & \cdots & e \\ x & x & x & e & e & \cdots & e \\ e & e & e & e & e & \cdots & e \\ e & e & e & e & e & \cdots & e \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ e & e & e & e & e & \cdots & e \end{bmatrix}}_{W} + \underbrace{\begin{bmatrix} x \\ x \\ x \\ x \\ \vdots \\ \vdots \\ x \end{bmatrix}}_{z} \underbrace{\begin{bmatrix} x & x & x & x & \cdots & \cdots & x \end{bmatrix}}_{z^H}, \tag{4.11}$$

where the upper left $r \times r$ elements of W, denoted with *x's*, will have very large values compared to the rest $\left(M - r\right) \times \left(M - r\right)$ of the elements, denoted with *e's*. Now that the matrix Ψ is defined, the next step in the algorithm is to find a Householder matrix P and apply a transformation to keep the first $r+1$ elements of z by clearing the last $M - r - 1$

elements. Householder matrices P (or elementary reflectors) [31] have the important ability to zero specified entries in a matrix or vector and are defined as

$$P = I - 2\frac{\nu\nu^T}{\nu^T\nu}, \tag{4.12}$$

with I the identity matrix,

$$\nu = z \pm \|z\|_2 \, e_1, \tag{4.13}$$

and

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{4.14}$$

so that

$$Pz = \pm \|z\|_2 \, e_1. \tag{4.15}$$

For example if $z = [2,\ 1,\ 4,\ 2]^T$ then $\|z\|_2 = 5,\ \nu = [7,\ 1,\ 4,\ 2]^T$ and

$$P = -\frac{1}{5}\begin{bmatrix} 2 & 1 & 4 & 2 \\ 1 & -34/7 & 4/7 & 2/7 \\ 4 & 4/7 & -19/7 & 8/7 \\ 2 & 2/7 & 8/7 & -31/7 \end{bmatrix},$$

such that

$$Pz = -5e_1 = [-5,\ 0,\ 0,\ 0]^T.$$

After computing the $(M-r)\times(M-r)$ Householder matrix P, needed to clear the last $M-r-1$ entries of vector z, we form the $M \times M$ block diagonal matrix $\tilde{P}$ as

$$\tilde{P} = \begin{bmatrix} I_r & 0_{r\times(M-r)} \\ 0_{(M-r)\times r} & P_{(M-r)\times(M-r)} \end{bmatrix}. \tag{4.16}$$

With the help of equation (4.16) the updated matrix $\tilde{\Psi}$ is given by

$$\tilde{\Psi} = \tilde{P}\Psi\tilde{P}^H = \tilde{P}\left(W + zz^H\right)\tilde{P}^H, \tag{4.17}$$

and is of the form

$$\tilde{\Psi} = \begin{bmatrix} x & x & x & x & e & \cdots & e \\ x & x & x & x & e & \cdots & e \\ x & x & x & x & e & \cdots & e \\ x & x & x & x & e & \cdots & e \\ e & e & e & e & e & \cdots & e \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ e & e & e & e & e & \cdots & e \end{bmatrix}. \tag{4.18}$$

The upper left $(r+1)\times(r+1)$ submatrix of $\tilde{\Psi}$ denoted with *x's* is unknown and, in order to proceed with the next cycle of the iterative algorithm, the rank $\tilde{r}$ of $\tilde{\Psi}$ needs to be estimated. This is done by computing the eigenvalues of the $(r+1)\times(r+1)$ block of *x's* and comparing the two smallest ones by putting a simple decision threshold. Specifically, if $\lambda_{r+1} \geq 0.1\lambda_r$ then we determine that the rank has changed and $\tilde{r} = r+1$. On the other hand if $\lambda_{r+1} < 0.1\lambda_r$ the rank of the updated matrix $\tilde{\Psi}$ remains unchanged and $\tilde{r} = r$. The way that the rank changes are determined in this implementation of the CRV updating algorithm is not the most efficient one as far as computations are concerned. For "r" small (say 2 or 3) there are closed formulas that can be applied for the computation of the eigenvalues, as the Lanczos bidiagonilization algorithm [10]. These formulas offer the same accuracy at less computational cost but are not examined in this research. All the steps described above are repeated each time a new signal (vector x) is received. At the end of each cycle, the correlation matrix $\tilde{\Phi}$ is computed from Equation (4.10) and is given by

$$\tilde{\Phi} = \tilde{V}\tilde{\Psi}\tilde{V}^H, \tag{4.19}$$

where

$$\tilde{V} = V\tilde{P}. \tag{4.20}$$

A graphical representation of the CRV updating algorithm is illustrated in the following flowchart (Figure 24).

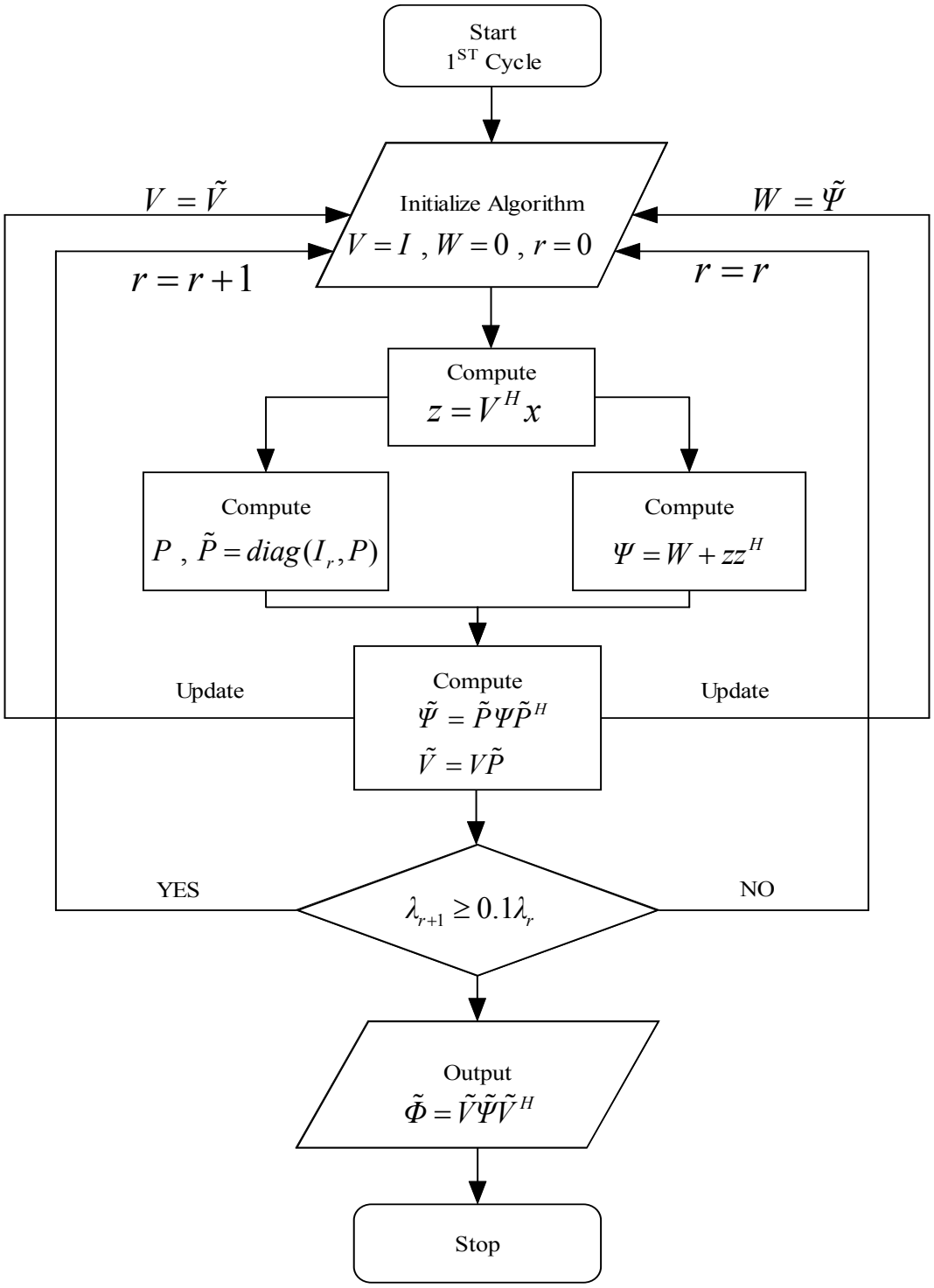

Figure 24. CRV Updating Algorithm.

C. NUMERICAL EXAMPLE

In this section a simple numerical example (done with MATLAB) is presented to demonstrate the accuracy of the correlation matrix estimate produced with the CRV updating algorithm illustrated in Figure 24. We start by creating a 10×10 data matrix *X*. Each column of *X* represents a random signal x_i, $i=1,2,....,10$ of ones and minus ones corrupted with additive white noise:

$$X=\begin{bmatrix} -0.315 & -0.781 & -2.837 & 1.506 & -2.731 & -0.006 & -0.223 & 0.654 & -0.199 & -1.379 \\ 0.698 & -1.66 & -0.502 & -2.028 & 1.478 & 0.306 & -0.696 & 1.986 & 1.88 & -1.242 \\ -1.599 & 2.351 & -0.18 & -0.9 & 0.552 & 2.199 & -1.73 & 1.643 & 0.775 & -0.165 \\ -0.219 & 1.136 & 0.765 & 0.833 & 1.386 & -3.577 & -2.143 & 1.919 & -0.703 & 1.756 \\ 1.602 & -0.833 & -0.631 & 1.689 & 1.053 & -1.086 & -0.413 & -2.248 & -1.521 & 2.164 \\ -0.057 & -0.705 & -1.317 & 0.883 & 0.513 & -0.613 & 0.408 & -0.842 & 0.844 & -2.023 \\ -0.023 & 1.276 & 0.203 & -0.674 & 1.244 & 0.138 & 1.518 & 1.788 & 0.901 & 0.701 \\ 0.932 & 1.394 & -0.309 & 0.904 & 1.718 & -2.23 & -2.492 & 0.422 & -0.002 & -1.494 \\ 1.081 & -1.098 & 0.957 & 1.031 & -0.846 & 1.641 & 0.913 & -0.472 & -0.565 & -0.827 \\ -2.767 & 1.176 & 1.324 & -1.613 & -0.866 & 0.095 & 0.987 & 2.671 & 0.974 & 1.354 \\ x_1 & x_2 & x_3 & ... & ... & ... & ... & ... & ... & x_{10} \end{bmatrix}. \quad (4.21)$$

The correlation matrix estimate is computed for the signals x_1 and x_2 by going through all the steps of the first two cycles of the algorithm.

1. First Cycle

The algorithm is initialized by setting $W=0$ and $V=I$. Both matrices are of dimension 10×10 and the rank *r* of *W* is zero. Next we need to compute the *z* vector given by Equation (4.9):

$$z=V^H x_1=I_{10}x_1=\begin{bmatrix} -0.315 \\ 0.698 \\ -1.599 \\ -0.219 \\ 1.602 \\ -0.057 \\ -0.023 \\ 0.932 \\ 1.081 \\ -2.767 \end{bmatrix}, \quad (4.22)$$

where x_1 is the first column of the data matrix X. From Equation (4.16) and the fact that $r = 0$ we get $\tilde{P} = P$, where P is the Householder matrix needed to clear (set to zero) all but the first element of z. The Householder matrix P is computed with the help of the expressions (4.12) through (4.14). Now that P is defined the updated matrix $\tilde{\Psi}$ given by (4.17) is

$$\tilde{\Psi} = \tilde{P}\Psi\tilde{P}^H = P\Psi P^H = P\left(W + zz^H\right)P^H \overset{W=0}{=} P\,zz^H P^H = \begin{bmatrix} 15.463 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{10\times 10} . \quad (4.23)$$

and the updated matrix $\tilde{V}$ by (4.20):

$$\tilde{V} = \begin{bmatrix} -0.08 & 0.177 & -0.406 & -0.055 & 0.407 & -0.014 & -0.006 & 0.237 & 0.275 & -0.703 \\ 0.177 & 0.97 & 0.066 & 0.009 & -0.067 & 0.002 & 0.001 & -0.038 & -0.045 & 0.115 \\ -0.406 & 0.066 & 0.846 & -0.021 & 0.153 & -0.005 & -0.002 & 0.089 & 0.103 & -0.264 \\ -0.055 & 0.009 & -0.021 & 0.997 & 0.021 & -0.0007 & -0.0003 & 0.012 & 0.014 & -0.036 \\ 0.407 & -0.067 & 0.153 & 0.021 & 0.846 & 0.005 & 0.002 & -0.089 & -0.103 & 0.265 \\ -0.014 & 0.002 & -0.005 & -0.0007 & 0.005 & 0.999 & 0.000 & 0.003 & 0.003 & -0.009 \\ -0.006 & 0.001 & -0.002 & -0.0003 & 0.002 & 0.000 & 0.999 & 0.001 & 0.001 & -0.003 \\ 0.237 & -0.038 & 0.089 & 0.012 & -0.089 & 0.003 & 0.001 & 0.947 & -0.06 & 0.154 \\ 0.275 & -0.045 & 0.103 & 0.014 & -0.103 & 0.003 & 0.001 & -0.06 & 0.929 & 0.179 \\ -0.703 & 0.115 & -0.264 & -0.036 & 0.265 & -0.009 & -0.003 & 0.154 & 0.179 & 0.541 \end{bmatrix} . \quad (4.24)$$

The last step is to compute the correlation matrix estimate $\tilde{\Phi}$ (4.19):

$$\tilde{\Phi} = \begin{bmatrix} 0.099 & -0.22 & 0.505 & 0.069 & -0.506 & 0.018 & 0.007 & -0.295 & -0.341 & 0.873 \\ -0.22 & 0.487 & -1.117 & -0.153 & 1.119 & -0.039 & -0.016 & 0.651 & 0.755 & -1.932 \\ 0.505 & -1.117 & 2.558 & 0.351 & -2.564 & 0.091 & 0.038 & -1.491 & -1.73 & 4.426 \\ 0.069 & -0.153 & 0.351 & 0.048 & -0.352 & 0.012 & 0.005 & -0.205 & -0.237 & 0.608 \\ -0.506 & 1.119 & -2.564 & -0.352 & 2.569 & -0.091 & -0.038 & 1.494 & 1.734 & -4.435 \\ 0.018 & -0.039 & 0.091 & 0.012 & -0.091 & 0.003 & 0.001 & -0.053 & -0.061 & 0.158 \\ 0.007 & -0.016 & 0.038 & 0.005 & -0.038 & 0.001 & 0.0005 & -0.022 & -0.025 & 0.066 \\ -0.294 & 0.651 & -1.491 & -0.205 & 1.494 & -0.053 & -0.022 & 0.868 & 1.008 & -2.579 \\ -0.341 & 0.755 & -1.73 & -0.237 & 1.734 & -0.061 & -0.025 & 1.008 & 1.17 & -2.993 \\ 0.873 & -1.932 & 4.426 & 0.608 & -4.435 & 0.158 & 0.066 & -2.579 & -2.993 & 7.656 \end{bmatrix} . \quad (4.25)$$

The rank r of $\tilde{\Psi}$ is determined (only in the first cycle of the algorithm) with the Matlab function "rank," which provides an estimate of the number of linearly independent rows or columns of a matrix. In this example the rank r of $\tilde{\Psi}$ is one.

2. Second Cycle

The CRV updating algorithm is a recursive one; thus the updated matrices $\tilde{V}$ and $\tilde{W} = \tilde{\Psi}$ computed in the first cycle are used to initialize the second cycle by setting $W = \tilde{W}$, $V = \tilde{V}$ and $r = 1$. The z vector is computed from (4.9):

$$z = V^H x_2 = \begin{bmatrix} -2.388 \\ -1.396 \\ 1.746 \\ 1.053 \\ -0.226 \\ -0.726 \\ 1.267 \\ 1.747 \\ -0.689 \\ 0.129 \end{bmatrix}, \tag{4.26}$$

where x_2 is the second column of X. The Householder matrix P needed to zero all except the first two $(r+1)$ elements of z is determined from Equations (4.12) through (4.14). For computational efficiency P is determined by operating on the last 9 $(10-r)$ elements of z and is of the dimension 9×9:

$$P = \begin{bmatrix} -0.405 & 0.507 & 0.306 & -0.065 & -0.211 & 0.368 & 0.507 & -0.200 & 0.037 \\ 0.507 & 0.816 & -0.110 & 0.023 & 0.076 & -0.132 & -0.183 & 0.072 & -0.013 \\ 0.306 & -0.110 & 0.933 & 0.014 & 0.046 & -0.080 & -0.110 & 0.043 & -0.008 \\ -0.065 & 0.023 & 0.014 & 0.996 & -0.009 & 0.017 & 0.023 & -0.009 & 0.001 \\ -0.211 & 0.076 & 0.046 & -0.009 & 0.968 & 0.055 & 0.076 & -0.030 & 0.005 \\ 0.368 & -0.132 & -0.080 & 0.017 & 0.055 & 0.903 & -0.133 & 0.052 & -0.009 \\ 0.507 & -0.183 & -0.110 & 0.023 & 0.076 & -0.133 & 0.816 & 0.072 & -0.013 \\ -0.200 & 0.072 & 0.043 & -0.009 & -0.030 & 0.052 & 0.072 & 0.971 & 0.005 \\ 0.037 & -0.013 & -0.008 & 0.001 & 0.005 & -0.009 & -0.013 & 0.005 & 0.998 \end{bmatrix}_{9\times 9}. \tag{4.27}$$

The block diagonal matrix $\tilde{P}$ is determined from Equation (4.16) as

$$\tilde{P} = \begin{bmatrix} I_r & 0_{r\times(M-r)} \\ 0_{(M-r)\times r} & P_{(M-r)\times(M-r)} \end{bmatrix} = \begin{bmatrix} I_1 & 0_{1\times 9} \\ 0_{9\times 1} & P_{9\times 9} \end{bmatrix}, \tag{4.28}$$

where P is the Householder matrix given by (4.27). The resulting matrix is

$$\tilde{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.405 & 0.507 & 0.306 & -0.065 & -0.211 & 0.368 & 0.507 & -0.200 & 0.037 \\ 0 & 0.507 & 0.816 & -0.110 & 0.023 & 0.076 & -0.132 & -0.183 & 0.072 & -0.013 \\ 0 & 0.306 & -0.110 & 0.933 & 0.014 & 0.046 & -0.080 & -0.110 & 0.043 & -0.008 \\ 0 & -0.065 & 0.023 & 0.014 & 0.996 & -0.009 & 0.017 & 0.023 & -0.009 & 0.001 \\ 0 & -0.211 & 0.076 & 0.046 & -0.009 & 0.968 & 0.055 & 0.076 & -0.030 & 0.005 \\ 0 & 0.368 & -0.132 & -0.080 & 0.017 & 0.055 & 0.903 & -0.133 & 0.052 & -0.009 \\ 0 & 0.507 & -0.183 & -0.110 & 0.023 & 0.076 & -0.133 & 0.816 & 0.072 & -0.013 \\ 0 & -0.200 & 0.072 & 0.043 & -0.009 & -0.030 & 0.052 & 0.072 & 0.971 & 0.005 \\ 0 & 0.037 & -0.013 & -0.008 & 0.001 & 0.005 & -0.009 & -0.013 & 0.005 & 0.998 \end{bmatrix}, \tag{4.29}$$

and the updated matrix $\tilde{\Psi} = \tilde{P}\Psi\tilde{P}^H = \tilde{P}\left(W + zz^H\right)\tilde{P}^H$ produces the following:

$$\tilde{\Psi} = \begin{bmatrix} 21.17 & -8.219 & 0 & \cdots & 0 \\ -8.219 & 11.838 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}_{10\times 10}, \tag{4.30}$$

where the entries of $\tilde{\Psi}$, denoted with zeros, are very small numbers $\left(<10^{-8}\right)$. The eigenvalues of the upper left 2×2 $\left(r+1\times r+1\right)$ block of $\tilde{\Psi}$ are $\lambda_1 = 25.955$ and $\lambda_2 = 7.052$ and, since $\lambda_2 > 0.1\lambda_1$, we determine that the rank has changed by one and is now $r = r+1 = 2$. It must be noted here that the decision threshold depends on the application, and there is no specific method known so far that could lead to an optimum choice [9]. The signal subspace power is concentrated in the upper left 2×2 block of $\tilde{\Psi}$ whereas the noise subspace is in the lower right 8×8 block. Finally, the correlation matrix estimate $\tilde{\Phi}$ calculated from Equations (4.19) and (4.20) is the following:

$$\tilde{\Phi} = \begin{bmatrix} 0.710 & 1.076 & -1.331 & -0.818 & 0.144 & 0.569 & -0.989 & -1.383 & 0.516 & -0.045 \\ 1.076 & 3.244 & -5.021 & -2.040 & 2.503 & 1.131 & -2.136 & -1.664 & 2.579 & -3.886 \\ -1.331 & -5.021 & 8.087 & 3.023 & -4.523 & -1.566 & 3.039 & 1.788 & -4.313 & 7.192 \\ -0.818 & -2.040 & 3.023 & 1.339 & -1.299 & -0.788 & 1.455 & 1.379 & -1.486 & 1.945 \\ 0.144 & 2.503 & -4.523 & -1.299 & 3.263 & 0.495 & -1.102 & 0.332 & 2.649 & -5.415 \\ 0.569 & 1.131 & -1.566 & -0.788 & 0.495 & 0.500 & -0.898 & -1.036 & 0.712 & -0.671 \\ -0.989 & -2.136 & 3.039 & 1.455 & -1.102 & -0.898 & 1.630 & 1.757 & -1.428 & 1.567 \\ -1.383 & -1.664 & 1.788 & 1.379 & 0.332 & -1.036 & 1.757 & 2.813 & -0.523 & -0.938 \\ 0.516 & 2.579 & -4.313 & -1.486 & 2.649 & 0.712 & -1.428 & -0.523 & 2.377 & -4.285 \\ -0.045 & -3.886 & 7.192 & 1.945 & -5.415 & -0.671 & 1.567 & -0.938 & -4.285 & 9.040 \end{bmatrix}. \quad (4.31)$$

The exact correlation matrices for the first two signals x_1 and x_2 are determined with the following expressions:

$$R_1 = x_1 x_1^H, \quad (4.32)$$

and

$$R_2 = R_1 + x_2 x_2^H. \quad (4.33)$$

After carrying out the calculations, the results from Equations (4.32) and (4.33) are exactly the same with the correlation matrix estimates (4.25) and (4.31) computed with the CRV updating algorithm. The eigenvalues of the exact correlation matrix and the eigenvalues of the estimated one for the 1st, 3rd, 6th and 10th iteration are illustrated in Figure 25. By looking at the figure, we conclude that the estimate produced by the CRV algorithm is very accurate.

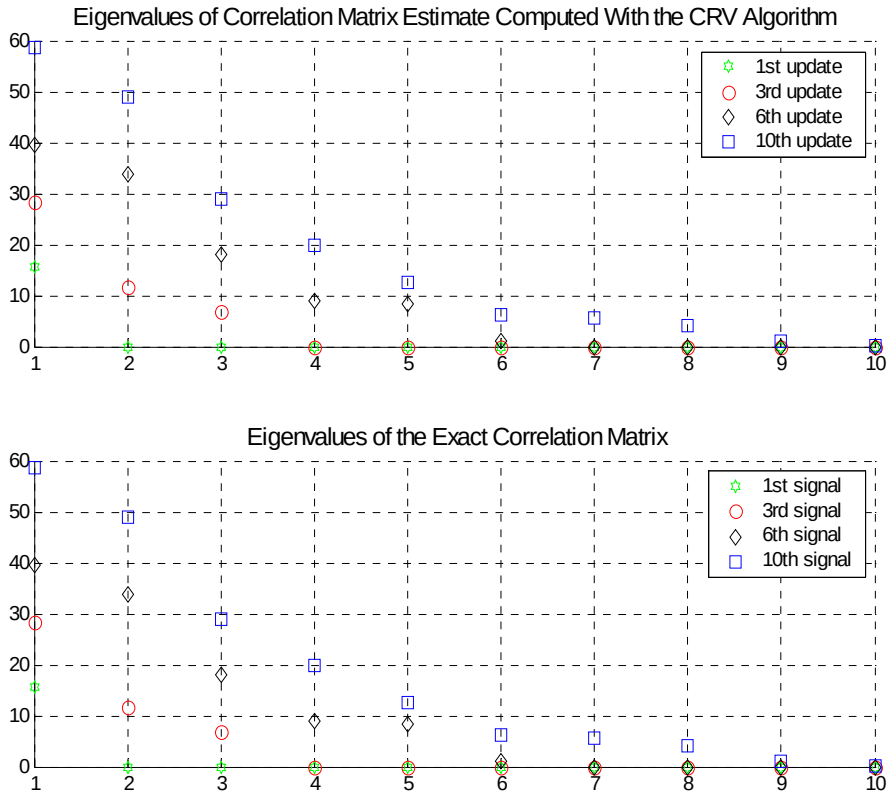

Figure 25. Values of the *n*-th Eigenvalue of the Estimated and the Exact Correlation Matrix.

In this chapter, the CRV updating algorithm [9] for subspace tracking was described and a simple numerical example was presented so that its accuracy could be evaluated. In the next chapter of this thesis the performance of the algorithm is examined by implementing it in a previously proposed communication scheme [2] under varying channel conditions.

V. IMPLEMENTATION AND PERFORMANCE OF THE CRV UPDATING ALGORITHM

This chapter describes the implementation of the CRV updating algorithm in a previously proposed direct sequence spread-spectrum (DS-SS) communication scheme [2]. The performance of the algorithm is evaluated under different channel conditions and design parameters (packet size, bit rate). The theory relevant to spread-spectrum (SS) signaling is presented first and the receiver structure is described next. The performance results are presented last in terms of Bit-Error-Rate (BER) vs. Bit Energy per Noise Power (E_b/N_o), as it is usually done in all communication systems.

A. DIRECT SEQUENCE SPREAD SPECTRUM (DS-SS)

Spread-spectrum (SS) techniques are preferred in military communications because of their low probability of intercept (LPI), low probability of detection (LPD) and their anti-jam capability [8]. In SS applications the resulting bandwidth of the transmitted signal is much greater than the information bandwidth and the transmitted signal appears to an unauthorized receiver as noise. The two most commonly used SS signaling schemes are frequency hopping (FH) and direct sequence (DS). Only DS is described in this thesis. The block diagram of a simple DS-SS system is illustrated in Figure 26.

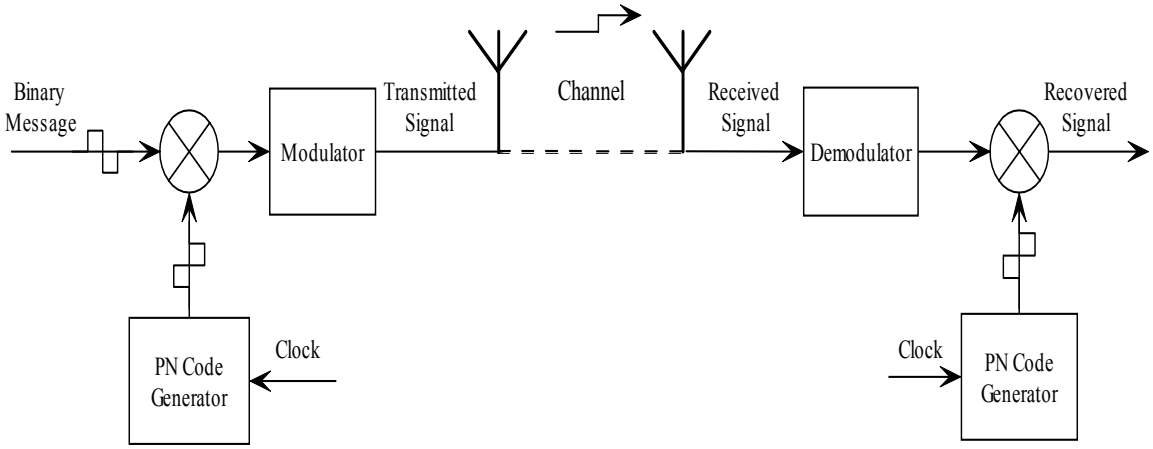

Figure 26. A Simple DS-SS System.

The binary message signal is spread at baseband by a locally generated pseudo-noise code (PN-code). The PN sequence $c(t)$, also known as the *chipping sequence*, is a

binary signal $\left(c(t) = \pm 1\right)$ that appears random and is produced at much higher frequency than the data that is to be transmitted. That is,

$$R_c >> R_b, \tag{4.34}$$

where R_c is the chip rate in chips-per-second [cps] and R_b is the information data rate in bits-per-second [bps]. Since $R_c >> R_b$ the chip duration T_c is much less than the bit duration T_b. The processing gain (or multiplicity factor) N (always an integer) in chips-per-bit [cpb] for a DS system is

$$N = \frac{R_c}{R_b} = \frac{T_b}{T_c}. \tag{4.35}$$

Therefore the message spectrum is spread N times before the message is modulated and transmitted. The wide bandwidth provided by the PN code allows the signal to be hidden in noise since its power is dropped below the noise threshold. In this way SS signals are very difficult to detect (LPD). The process of spreading by multiplication with a PN sequence is shown in Figure 27 for the case of $N = 2$ $(T_c = 0.5T_b)$.

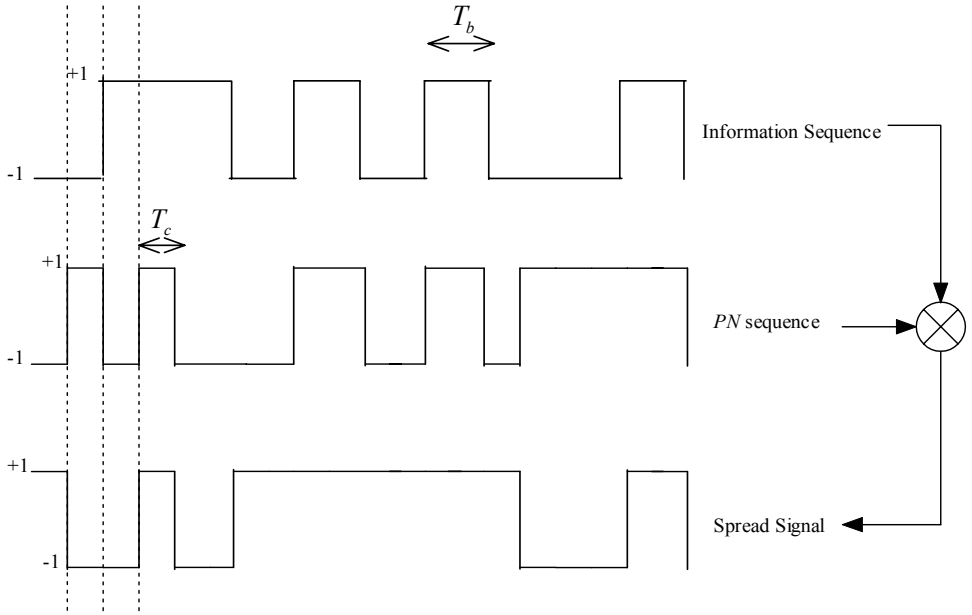

Figure 27. Example of Spreading in a DS-SS System.

There are many different PN codes used in DS-SS communications like the maximal length codes, the Hadamard-Walsh codes, the Gold codes, the Kasami codes, and others. In the implementation of this thesis research, the spreading of the data signal is done by Gold codes, which are produced with the Gold code generators of SIMULINK. Gold codes have very good cross-correlation properties (the cross-correlation of two different Gold codes is very small) and are preferred when multiple users must share the same frequency spectrum (each user is assigned a different code). As shown in Figure 21, the exact replica of the PN code used at the transmitter is needed in order for the receiver to decode the transmitted signal. The decoding is done by a simple multiplication (de-spreading) of the received signal with the chipping code since $c^2(t) = 1 \left(c(t) = \pm 1 \right)$. The fact that the receiver must be able to generate the same spreading sequences in a deterministic way, just as the transmitter does, provides a certain degree of communication security. The transmitted signal if detected by a non-intended receiver cannot be intercepted (decoded) without prior knowledge of the specific PN sequence used for the spreading of the information signal (LPI).

The third important advantage in using SS signaling is the ability to resist hostile interference (jamming). A useful parameter in specifying the performance of a spread-spectrum system in the presence of jamming is the processing gain *N*. Since the message signal is spread before transmission, the interference bandwidth occupies only a small portion of the SS signal bandwidth, as shown in Figure 28 (left plot). At the receiver, after de-spreading the received signal by multiplying with the same PN code (the one used at the transmitter), the spectrum of Figure 28 (right plot) is produced. Most of the original interference energy is eliminated by spreading (except for the energy corresponding to the interference spectrum overlapping with the signal spectrum) and the original signal can be recovered. It is also clear from the same figure that the greater the processing gain *N* of the SS system, the better is the performance of the system against interference (barrage noise jamming) [18].

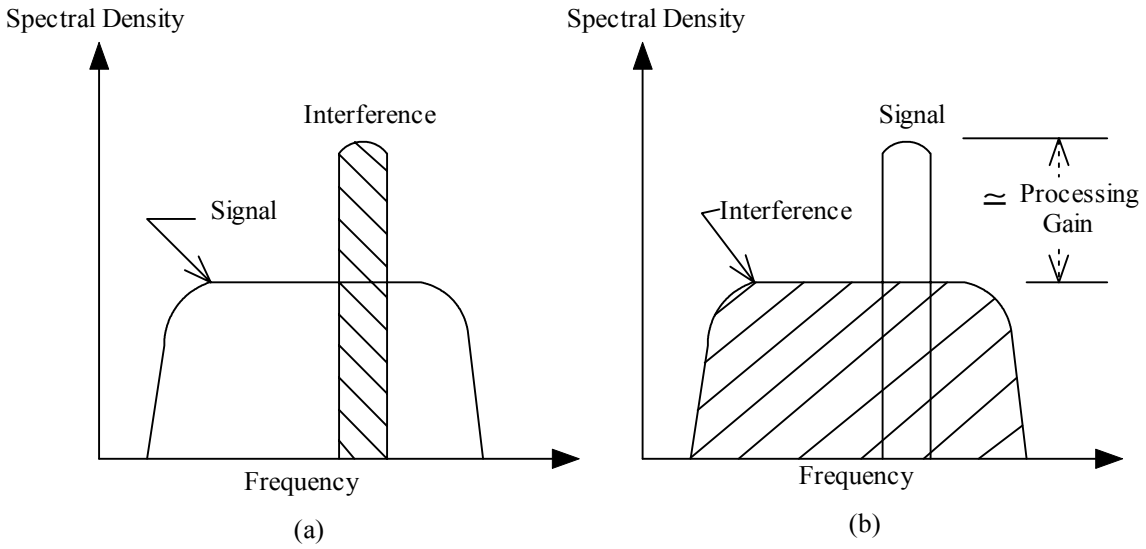

Figure 28. Spectrum of the Transmitted SS Signal (left plot) and of the Received Signal (right plot) after De-spreading [From Ref. 18].

B. PREVIOUSLY PROPOSED DS-SS SCHEME

The channel equalization proposed by [2] is based on multi-rate signal processing and subspace decomposition. The analysis is done in the discrete time domain and the transmitted signal is assumed to be a binary sequence (+1, -1), which is then chipped by a short PN code and transmitted through the channel. Also, error correction coding and modulation were not examined and are not implemented in this thesis. Thus the worst-case scenario is simulated since both processes (error correction coding and modulation) can only improve the equalization mechanism. The block diagram of the simplified DS-SS system is shown in Figure 29.

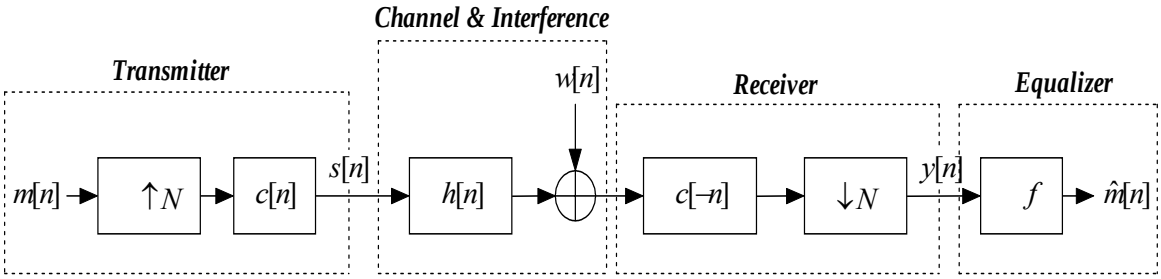

Figure 29. Simplified DS-SS System [After Ref. 2].

The information sequence $m[n]$ is first up-sampled by N (add $N-1$ zeros between bits) in order to match the chipping rate R_c and then is spread by convolution with

the chipping sequence $c[n]$. Spreading can also be done by multiplying the up-sampled (where each bit is repeated N times) data sequence $m[n]$ with $c[n]$. The spread signal $s[n]$ is then convolved with the channel impulse response $h[n]$ and noise $w[n]$ is added (white or colored) to account for the random interference. Subsequently, the received sequence is de-spread at the receiver by convolution with a time-reversed replica $c[-n]$ of the PN-code $c[n]$ and then is down-sampled by N. In order to effectively remove the "pseudo-randomization" imposed on the data sequence by the spreading code, a convolution between $c[n]$ and $c[-n]$ must yield a delta function,

$$c[n] * c[-n] = \delta[n]. \tag{4.36}$$

If $c[n] = [c_0, c_1, ..., c_{N-1}]$, where $c_0, c_1, ..., c_{N-1}$ are the chips of the PN-code, then

$$c[-n] = [c_{N-1}, ..., c_1, c_0]. \tag{4.37}$$

For example, when a Gold code $c[n]$ of length 125 is used and $c[-n]$ is given by Equation (4.37), the result of (4.36) is the impulse shown in Figure 30.

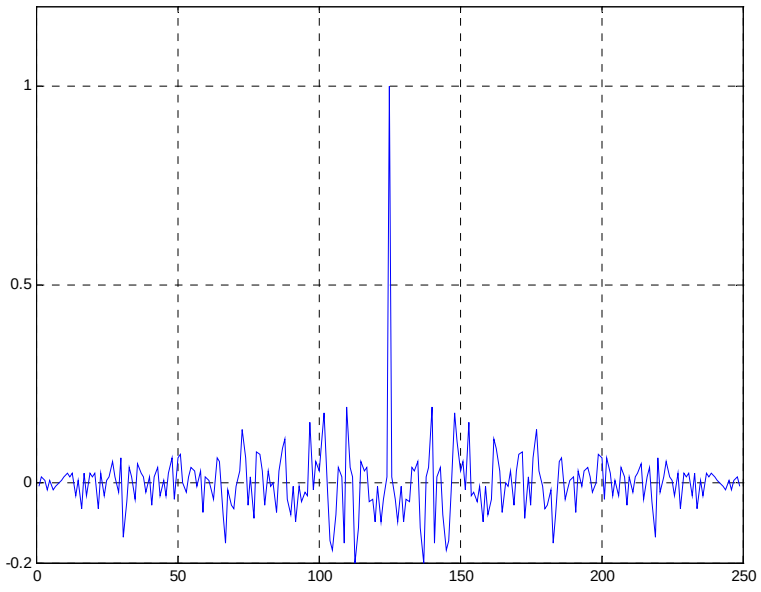

Figure 30. The Result of $c[n] * c[-n]$ When Gold Codes are Used.

The equivalent state-space structure of the communication system shown in Figure 29 is illustrated in Figure 31.

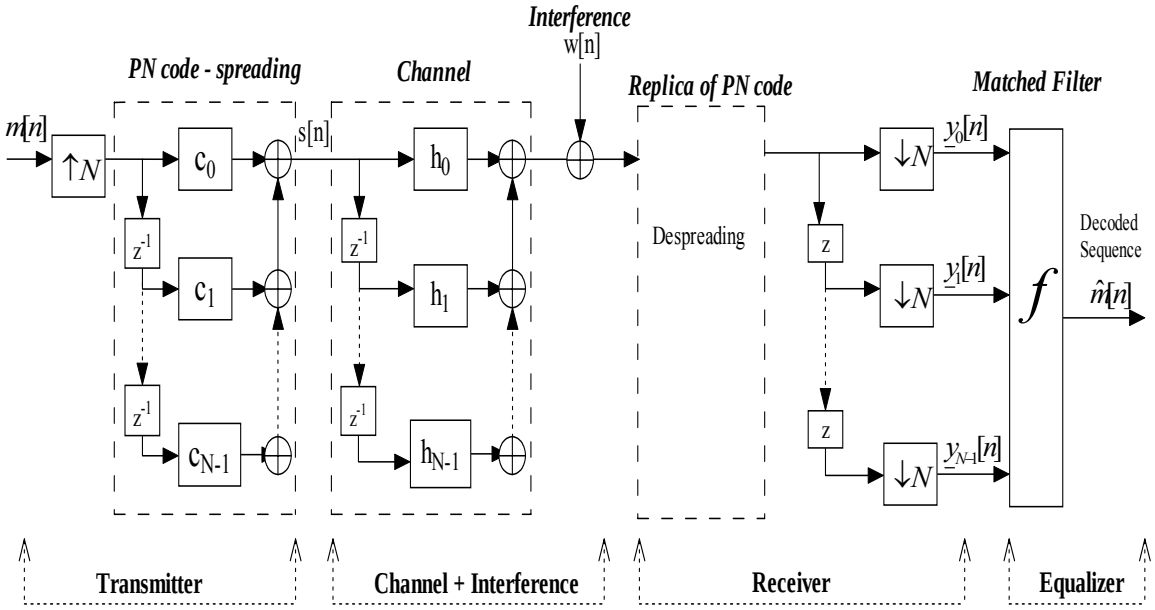

Figure 31. State-Space Structure of the DS-SS System [After Ref. 2].

The spreading of the information sequence $m[n]$ is accomplished by filtering the up-sampled (by N) data sequence by a N-th order linear filter defined (in the z-domain) as

$$C(z) = c_0 + c_1 z^{-1} + c_2 z^{-2} + ... + c_{N-1} z^{-(N-1)}. \tag{4.38}$$

The spread signal $s[n]$ is then filtered by a second N-th order linear filter (channel impulse response),

$$H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + ... + h_{N-1} z^{-(N-1)}, \tag{4.39}$$

to account for the corruption caused by the underwater acoustic channel. At the receiver the received sequence is de-spread by multiplication (and not convolution) with a replica of the PN-code used at the transmitter. In practice the length of the received sequence does not change after de-spreading. The summation step implied by the convolution approach $\left(c[n] * c[-n]\right)$ is performed later at the equalizer when the estimate $\hat{m}[n]$ of $m[n]$ is computed. At the equalizer (see Figure 31), the de-spread sequence is forward delayed (z elements in the state-space structure) and down-sampled by N. The result of this process is N different versions of the original information sequence (corrupted by the fading channel) denoted as $\underline{y}_0[n]$, $\underline{y}_1[n]$,....., $\underline{y}_{N-1}[n]$. For a better understanding of how the structure of Figure 31 works, simple examples are presented in Figure 32 (transmitter section) and Figure 33 (channel and receiver section).

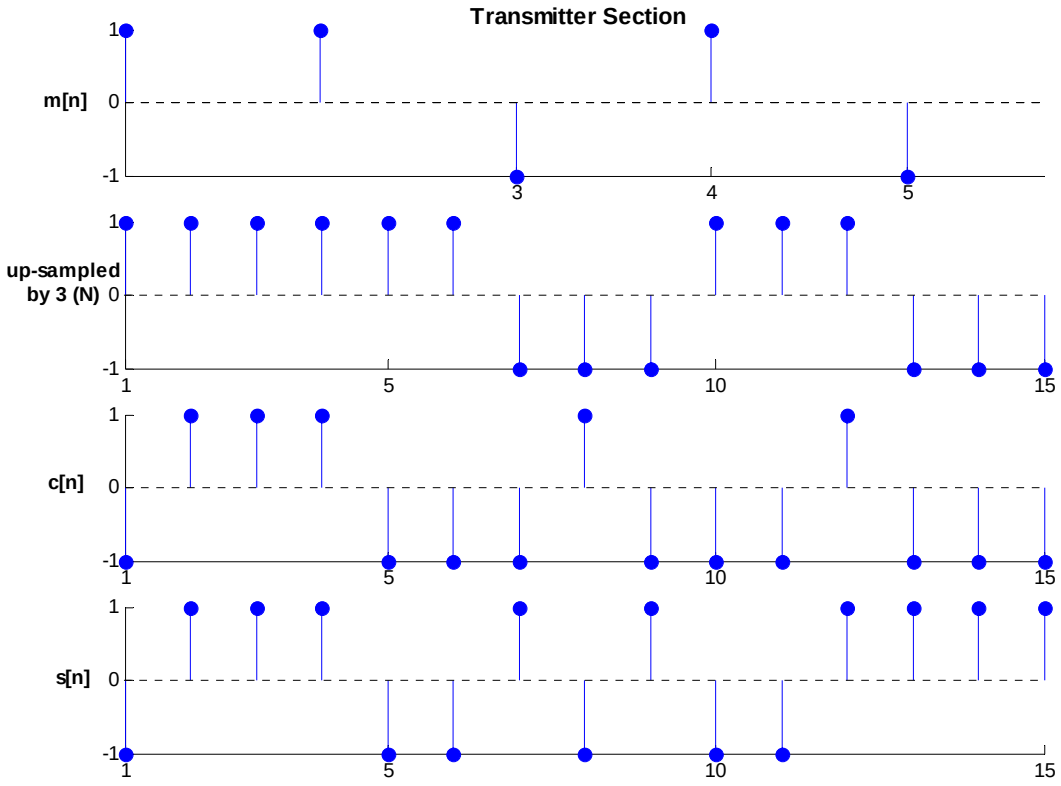

Figure 32. Up-sampling and Spreading by Multiplication with the PN-code.

The data sequence $m[n]$ consists of 5 bits $[1, 1, -1, 1, -1]$ and the processing gain is assumed to be three $(N = 3)$. Since $N = 3$, $m[n]$ is up-sampled by 3 (each bit is repeated three times) and then bit-by-bit multiplied with the PN-code $c[n]$ to yield the spread sequence $s[n]$. Subsequently, white noise $w[n]$ is added to account for the channel corruption. The sequence $s[n] + w[n]$ is then de-spread at the receiver by multiplication with a replica of $c[n]$ (Figure 33, third plot from top). Finally, the de-spread sequence is forward delayed and down-sampled by N, so that three different versions $y_0[n]$, $y_1[n]$, $y_2[n]$ of the original data sequence are produced at the equalizer input. The three sequences $y_0[n]$, $y_1[n]$, $y_2[n]$ are shown in Figure 33 (first plot from bottom) in different colors (blue, red, and green respectively).

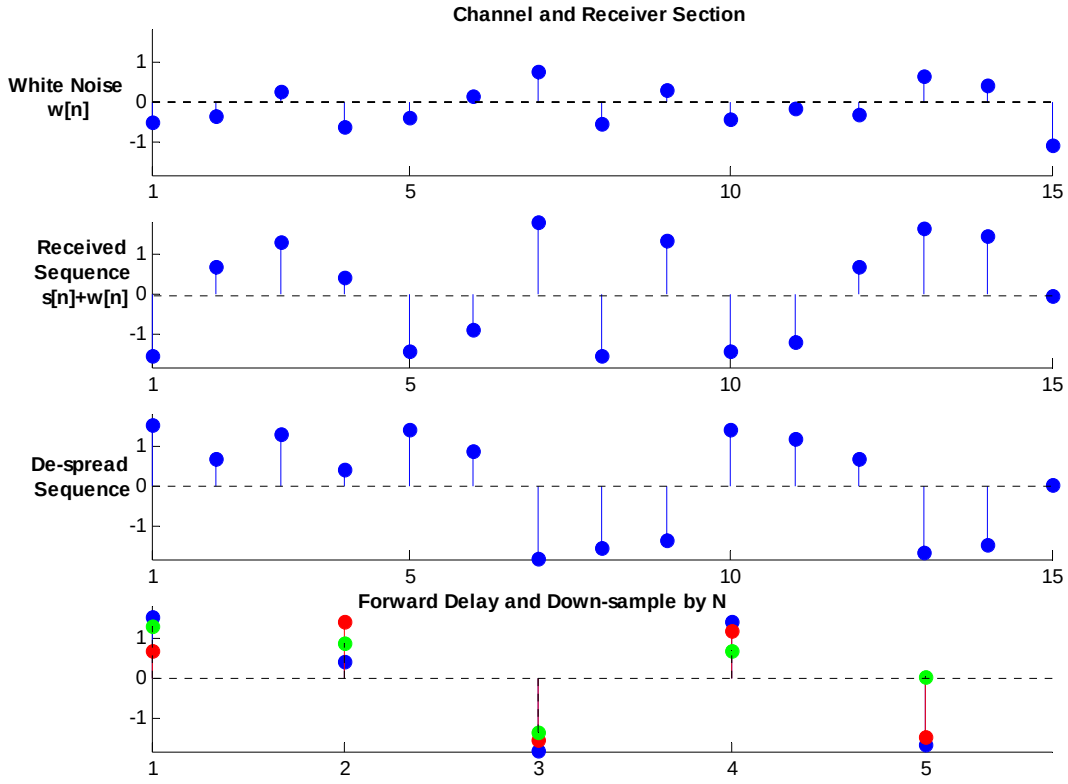

Figure 33. Channel Corruption, De-spreading and Different Versions of the Original Data Sequence.

The vectors $\underline{y}_0[n],\ \underline{y}_1[n],\ldots\ldots,\ \underline{y}_{N-1}[n]$ (see Figure 31) form the data matrix

$$y[n]=\begin{bmatrix} - & \underline{y}_0[n] & - \\ - & \underline{y}_1[n] & - \\ \vdots & \vdots & \vdots \\ - & \underline{y}_{N-1}[n] & - \end{bmatrix}, \tag{4.40}$$

which is then used to determine the signal and noise subspace (the most important part in the equalization process). The estimated sequence $\hat{m}[n]$ is given by

$$\hat{m}[n]=f^T\cdot y[n], \tag{4.41}$$

where f is the optimum filter that maximizes the signal-to-noise ratio at the receiver. The channel equalization is performed with two different blind equalization algorithms, depending on whether the synchronization between the receiver and the transmitter was

achieved or not. The two algorithms are based on matched filter theory and are briefly described in the following subsections. A detailed derivation and description can be found in Reference [2].

1. Demodulation with Synchronization

This is the case where the replica of the PN-code used at the transmitter is completely aligned (in time) with the received sequence. Thus the de-spreading is performed correctly.

a. Multipath with Additive White Gaussian Noise (AWGN)

The received sequence $y[n]$ is equal to $s[n]*h[n]+w[n]$ where $s[n]$ is the spread by the PN-code information sequence, $w[n]$ is the AWGN, and $h[n]$ is the channel impulse response. It must be noted here that the maximum length of the impulse response $h[n]$, according to the theoretical approach described in the previous section, is N, where N is the processing gain (see Figure 31). If $\hat{m}[n]$ is an estimate of the information sequence $m[n]$ then

$$\hat{m}[n] = f^T \cdot y[n], \tag{4.42}$$

where f is the eigenvector corresponding to the maximum eigenvalue of the data correlation matrix R_{yy} given by

$$R_{yy} = E\left\{y[n]y^T[n]\right\}. \tag{4.43}$$

b. Multipath with Additive Colored Noise (ACN)

The received sequence $y[n]$ is equal to $s[n]*h[n]+v[n]$ where $v[n]$ is the additive colored noise. In order to be able to follow the approach mentioned above in (4.42), a simultaneous diagonalization of the correlation matrices of the signal R_{ss} and the noise R_{vv} is needed. This is done with help of the Mahalanobis transformation, defined as [32]

$$y'[n] = M^{-1/2} y[n], \tag{4.44}$$

where $M^{-1/2}$ is the transformation matrix given by

$$M^{-1/2} = E_v \Lambda_v^{-1/2} E_v^T = R_{vv}^{-1/2}, \tag{4.45}$$

and E_v, Λ_v are the matrices produced from the eigenvalue decomposition of $R_{vv} = E\left\{v[n]v^T[n]\right\}$. Specifically E_v is a matrix with columns of the eigenvectors of R_{vv} and Λ_v is a diagonal matrix whose elements are the corresponding eigenvalues. The correlation matrix R'_{yy} of $y'[n]$ (4.44) is computed from

$$R'_{yy} = E\left\{y'[n]\left(y'[n]\right)^T\right\} = E\left\{\left(M^{-1/2}y[n]\right)\left(M^{-1/2}y[n]\right)^T\right\} = M^{-1/2}R_{yy}M^{-1/2}, \tag{4.46}$$

and, since $M^{-1/2} = R_{vv}^{-1/2}$, from

$$R'_{yy} = R_{vv}^{-1/2}R_{yy}R_{vv}^{-1/2}. \tag{4.47}$$

Because of this whitening transformation in the new coordinate system the colored noise is considered white and the estimate $\hat{m}[n]$ of the data signal $m[n]$ is given by (4.42)

$$\hat{m}[n] = \left(f'\right)^T y'[n], \tag{4.48}$$

where f' is the eigenvector corresponding to the maximum eigenvalue of the autocorrelation matrix R'_{yy}.

2. Demodulation without Synchronization

The underwater communication channel is very challenging and synchronization is not always possible. The slightest misalignment in the PN-code can incorrectly despread the received sequence. When synchronization is achieved, the data correlation matrix R_{yy} (4.43) has only one dominant eigenvalue and the signal subspace is easily discriminated from the noise subspace. However, this is not the case when synchronization is not possible. Suppose that, at a given time, $\underline{y}[n]$ contains two symbols, the current symbol $m[n]$ and a portion of the previous symbol $m[n-1]$, i.e.

$$\underline{y}[n] = \underline{g}_0 m[n] + \underline{g}_1 m[n-1] = \begin{bmatrix}\underline{g}_0 & \underline{g}_1\end{bmatrix}\begin{bmatrix} m[n] \\ m[n-1]\end{bmatrix}, \tag{4.49}$$

where $\underline{g}_0$, $\underline{g}_1$ are vectors depending on the autocorrelation of the PN-code and the communication channel. In this case $\underline{y}[n]$ spans a two-dimensional subspace defined by the vectors $\underline{g}_0$ and $\underline{g}_1$ and there are two dominant eigenvalues instead of one in the data correlation matrix R_{yy}. If $\lambda_1[n]$, $\lambda_2[n]$ are the two dominant eigenvalues and $\underline{e}_1$, $\underline{e}_2$ their respective eigenvectors, $\underline{y}[n]$ is given by:

$$\underline{y}[n] = \underline{e}_1\lambda_1[n] + \underline{e}_2\lambda_2[n]. \tag{4.50}$$

Since the eigenvectors are orthonormal ($\underline{e}_1^{\mathrm{T}}\underline{e}_1 = 1$ and $\underline{e}_1^{\mathrm{T}}\underline{e}_2 = 0$), it is true that

$$\lambda_1[n] = \underline{e}_1^T \cdot \underline{y}[n] \tag{4.51}$$

and

$$\lambda_2[n] = \underline{e}_2^T \cdot \underline{y}[n]. \tag{4.52}$$

By combining Equations (4.51) and (4.52) in matrix form, we get

$$\begin{bmatrix}\lambda_1[n]\\ \lambda_2[n]\end{bmatrix} = \begin{bmatrix}\underline{e}_1^T\\ \underline{e}_2^T\end{bmatrix} \cdot y[n] \Leftrightarrow \underline{\lambda}[n] = \begin{bmatrix}\underline{e}_1^T\\ \underline{e}_2^T\end{bmatrix} \cdot y[n]. \tag{4.53}$$

Substituting (4.49) into (4.53) yields

$$\underline{\lambda}[n] = \begin{bmatrix}\underline{e}_1^T\\ \underline{e}_2^T\end{bmatrix} \cdot \begin{bmatrix}\underline{g}_0 & \underline{g}_1\end{bmatrix} \cdot \begin{bmatrix}m[n]\\ m[n-1]\end{bmatrix}. \tag{4.54}$$

If we solve Equation (4.54) for $\begin{bmatrix}m[n]\\ m[n-1]\end{bmatrix}$, we get

$$\begin{bmatrix}m[n]\\ m[n-1]\end{bmatrix} = \left(\begin{bmatrix}\underline{e}_1^T\\ \underline{e}_2^T\end{bmatrix} \cdot \begin{bmatrix}\underline{g}_0 & \underline{g}_1\end{bmatrix}\right)^{-1} \cdot \underline{\lambda}[n] \Leftrightarrow \begin{bmatrix}m[n]\\ m[n-1]\end{bmatrix} = \begin{bmatrix}\underline{k}_1^T\\ \underline{k}_2^T\end{bmatrix} \cdot \underline{\lambda}[n], \tag{4.55}$$

where we assumed that $\underline{g}_0$, $\underline{g}_1$ (and $\underline{k}_1^T$, $\underline{k}_2^T$) are linearly independent, and

$$\begin{bmatrix}\underline{k}_1^T\\ \underline{k}_2^T\end{bmatrix} = \left(\begin{bmatrix}\underline{e}_1^T\\ \underline{e}_2^T\end{bmatrix} \cdot \begin{bmatrix}\underline{g}_0 & \underline{g}_1\end{bmatrix}\right)^{-1}. \tag{4.56}$$

From (4.55) the two information symbols are given by

$$m[n] = \underline{k}_1^T \cdot \underline{\lambda}[n], \tag{4.57}$$

and

$$m[n-1] = \underline{k}_2^T \cdot \underline{\lambda}[n] \Leftrightarrow m[n] = \underline{k}_2^T \cdot \underline{\lambda}[n+1]. \tag{4.58}$$

A combination of (4.57) and (4.58) yields

$$\underline{k}_1^T \cdot \underline{\lambda}[n] = \underline{k}_2^T \cdot \underline{\lambda}[n+1] \Leftrightarrow \underline{k}_1^T \cdot \underline{\lambda}[n] - \underline{k}_2^T \cdot \underline{\lambda}[n+1] = 0, \tag{4.59}$$

and in matrix form

$$\begin{bmatrix} \underline{k}_1^T & -\underline{k}_2^T \end{bmatrix} \cdot \begin{bmatrix} \underline{\lambda}[n] \\ \underline{\lambda}[n+1] \end{bmatrix} = 0. \tag{4.60}$$

In order to calculate $\underline{k} = \begin{bmatrix} \underline{k}_1 \\ -\underline{k}_2 \end{bmatrix}$, we start by calculating the autocorrelation matrix $R_{\lambda\lambda} = \sum_n \begin{bmatrix} \underline{\lambda}[n] \\ \underline{\lambda}[n+1] \end{bmatrix} \cdot \begin{bmatrix} \underline{\lambda}^T[n] & \underline{\lambda}^T[n+1] \end{bmatrix}$ with help of Equation (4.53). From (4.60) it is obvious that

$$\begin{bmatrix} \underline{k}_1^T & -\underline{k}_2^T \end{bmatrix} \cdot R_{\lambda\lambda} = 0. \tag{4.61}$$

Therefore, $\underline{k}$ is equal to the eigenvector corresponding to the minimum eigenvalue of $R_{\lambda\lambda}$. Once the estimate for $\underline{k}$ is known, the estimated sequence $\hat{m}[n]$ is derived from (4.55). The equalization algorithm mentioned above is tailored for the specific case that, at any given time, $\underline{y}[n]$ contains only two symbols (slight misalignment). If this is not the case and $\underline{y}[n]$ contains more than two symbols, the same approach can be followed (in this case there are more than two dominant eigenvalues) that would lead to a similar algorithm.

C. PROPOSED DS-SS SCHEME

The combination of the DS-SS modulation scheme and the receiver structure based on the aforementioned blind equalization algorithms was proven very robust as part of a previous thesis [2]. However, there is a major problem associated with this scheme. The desired subspaces can only be estimated offline since the whole received packet is needed in order to calculate the data correlation matrix R_{yy}. This places a computational burden that increases as the packet length increases.

In this thesis the signal and noise subspaces are estimated online (recursively) with help of the CRV updating algorithm presented in Chapter IV. By following this approach, the desired subspaces are efficiently estimated in a recursive manner. This is done by feeding each column of the received data matrix $y[n]$ (4.40) into the CRV algorithm. The columns of $y[n]$ represent the received spread bits, corresponding to one information bit spread by the PN-code and corrupted by the fading channel.

In the previously proposed communication scheme, the channel impulse response was generated using the *Bellhop* underwater acoustic propagation model [5]. The time variability of the channel was introduced to the static impulse response by allowing each of the multipath components to vary randomly within an arbitrary range. The random phase shift caused by the micro-paths was approximated by adding a random phase factor to each of the amplitudes.

In this thesis the channel impulse response is modeled using the Monterey-Miami parabolic equation model (MMPE) [7] described in Chapter III. The broadband impulse response is generated for the frequency range $11,500 \pm 1024\,\text{Hz}$ $(\text{BW} = 2048\,\text{Hz})$ under various perturbations including the influence of Doppler due to source motion. The worst-case scenario is examined by evaluating the CRV algorithm using a time-variable impulse response that also accounts for the interface roughness, the turbulence, the internal waves, and the volume scattering commonly found in shallow water environments.

D. SIMULATION RESULTS

The performance of the CRV updating algorithm evaluated under different channel conditions and design parameters (packet size, bit rate, etc.) is examined next. The DS-SS system structure of Figure 31 was implemented in MATLAB and the simulation results are presented in terms of Bit-Error-Rate (BER) vs. Bit Energy per Noise Power (E_b/N_o). Each simulation involved sending an adequate number of packets for each value of E_b/N_0 in order to create sufficient statistics. The E_b/N_o is a scaled version of the signal-to-noise ratio (SNR) defined as

$$\frac{E_b}{N_0} = \text{SNR} \cdot \frac{BW}{R_b}, \tag{4.62}$$

where BW is the available communication bandwidth and R_b is the bit rate. The range of the desired E_b/N_o used for the various simulations was between 0 and 30 dB. The BER (probability of a bit error) is a metric used to describe the reliability of a digital communication system quantitatively and is defined as

$$\text{BER}=\frac{\text{Total Number of Errors}}{\text{Total Number of Bits Sent}}. \tag{4.63}$$

1. Performance in AWGN

The received sequence is $s[n]+w[n]$ where $s[n]$ is the spread information sequence and $w[n]$ is the AWGN. In this case the channel corruption is not taken into account; thus, the results of this simulation can be treated as a benchmark for evaluating the receiver performance. Figure 34 shows the resulting BER for different packet sizes and Figure 35 the BER for various data rates R_b.

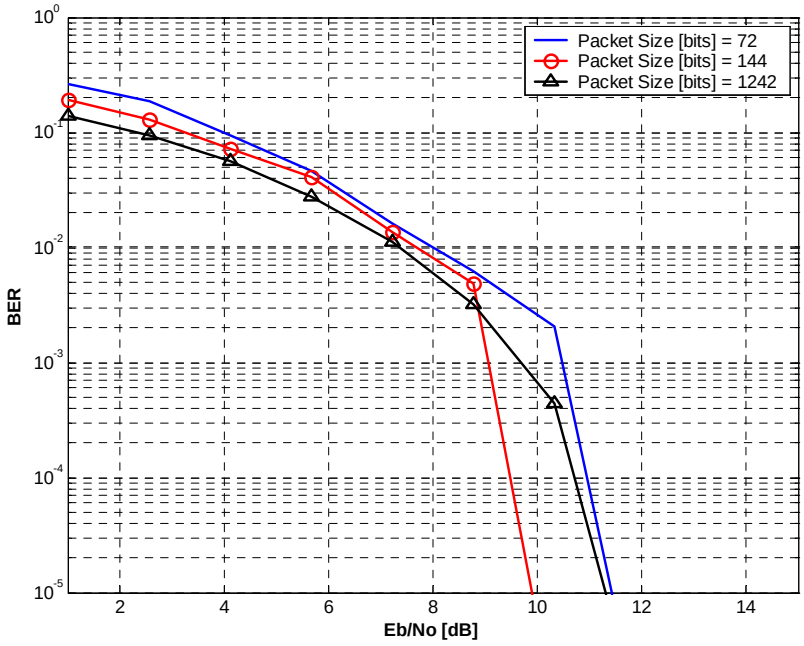

Figure 34. BER in an AWGN Channel When Varying the Packet Size $\left(R_b = 40\,\text{bps},\ R_c = 2400\,\text{cps}\right)$.

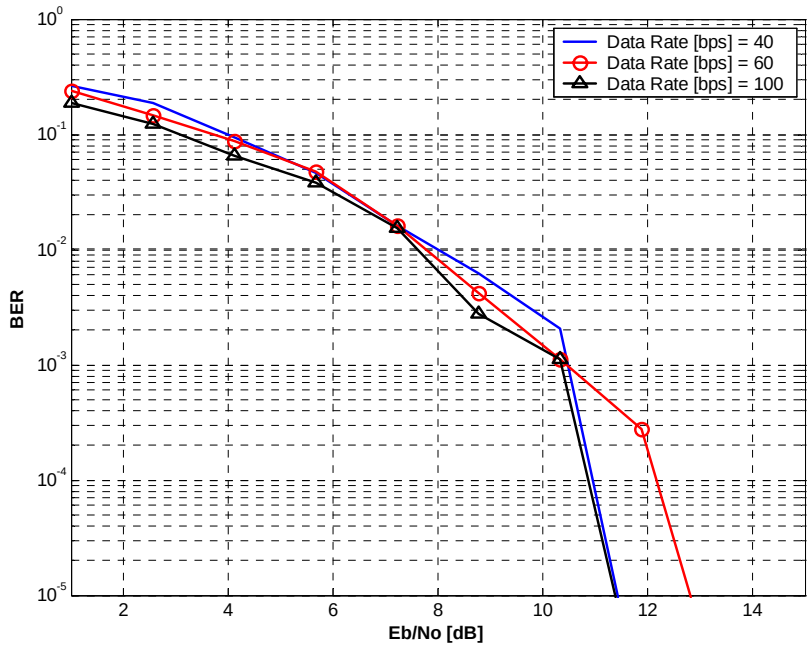

Figure 35. BER in an AWGN Channel When Varying the Data Rate R_b $\left(\text{packet size} = 72\,\text{bits},\ R_c = 2400\,\text{cps}\right)$.

From the results presented above two observations can be made. First, we see that, if longer transmission packets are used, the performance is not degraded in this ideal case. The same conclusion stands for the case where a higher data rate is used. The performance seems to be better for a packet of 144 bits (Figure 34) and for a data rate of 60 bps (Figure 35). This is possibly because of statistical variations.

2. Performance in Additive Color Noise

The colored noise is produced by filtering white noise by the 7th-order FIR filter given by

$$Q(z) = \frac{1}{z^6 - 0.9z^5 + 0.4z^4 - 0.2z^3 + 0.1z^2 - 0.1z}. \tag{4.64}$$

This is the same filter used in [2] so that the performance results may be compared.

Figure 36 illustrates the performance of the new subspace decomposition algorithm (CRV) in additive colored noise when varying the packet size. The results of this simulation are worse than the white noise case by approximately 4 dB. Figure 37 demonstrates the receiver performance for a 72-bit packet when the bit rate is varied in the range

of 40 to 100 bps. It is shown once again that the algorithm is stable even for a bit rate of 100 bps.

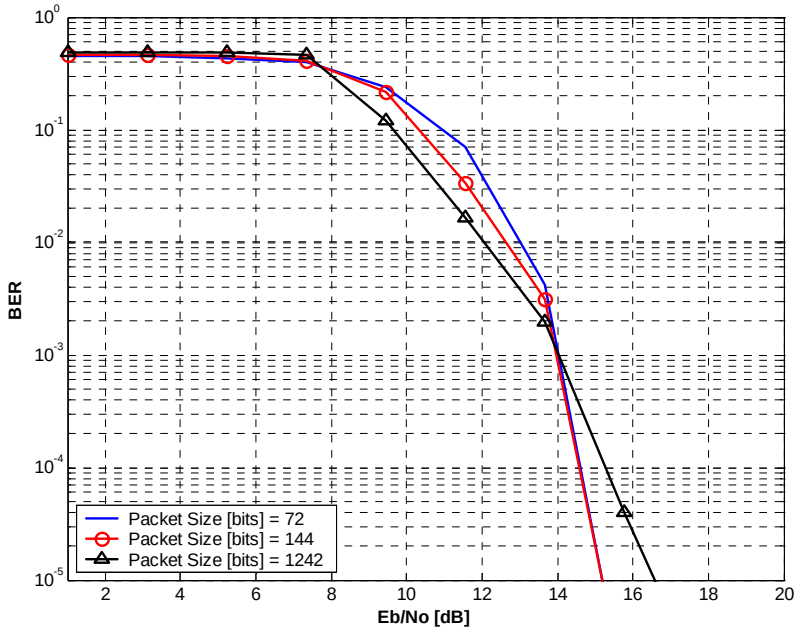

Figure 36. Performance in Additive Color Noise When Varying the Packet Size $(R_b = 40\,\text{bps},\ R_c = 2400\,\text{cps})$.

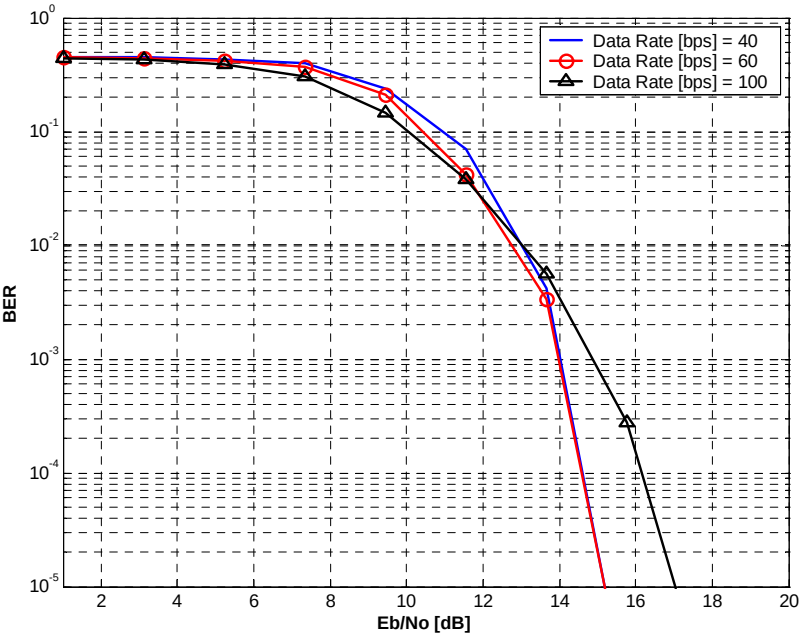

Figure 37. Performance in Additive Color Noise When Varying the Data Rate $(\text{packet size} = 72\,\text{bits},\ R_c = 2400\,\text{cps})$.

3. Performance in Multipath and AWGN

The received sequence is $s[n]*h[n]+w[n]$ where $s[n]$ is the spread by the PN-code data sequence, $w[n]$ is the AWGN, and $h[n]$ is the channel impulse response. The impulse responses used for the simulations are the ones computed with help of the MMPE model. The performance of the CRV algorithm was evaluated for all the different perturbations described in Chapter III and the results are shown in Figure 38 through Figure 44.

As already mentioned, the communication scheme of Figure 31 is tailored for a channel impulse response equal to the processing gain *N*. For all the following simulations, $R_b = 40\,\text{bps}$ and $R_c = 2400\,\text{cps}$, thus *N* is 60. This is the reason all results are presented for a channel length of at least 60 time samples.

Among the individual perturbations, the most challenging are those corresponding to the interface roughness (Figure 39) and turbulence (Figure 42). In those cases the results are good only for a channel length of 60 time samples.

When all the examined perturbations are combined (Figure 43) the results are poor and on the average worse than the previous cases by 18 dB (for an impulse response of 60 samples). The results of the worst-case scenario where the Doppler is also taken into account are presented in Figure 44. In this case, equalization cannot be performed, even for a channel length of 60 time samples. However, it must be noted that the examined perturbations are expected to be realistic but strong perturbations. It is doubtful that all of them combined will be encountered in many shallow water environments.

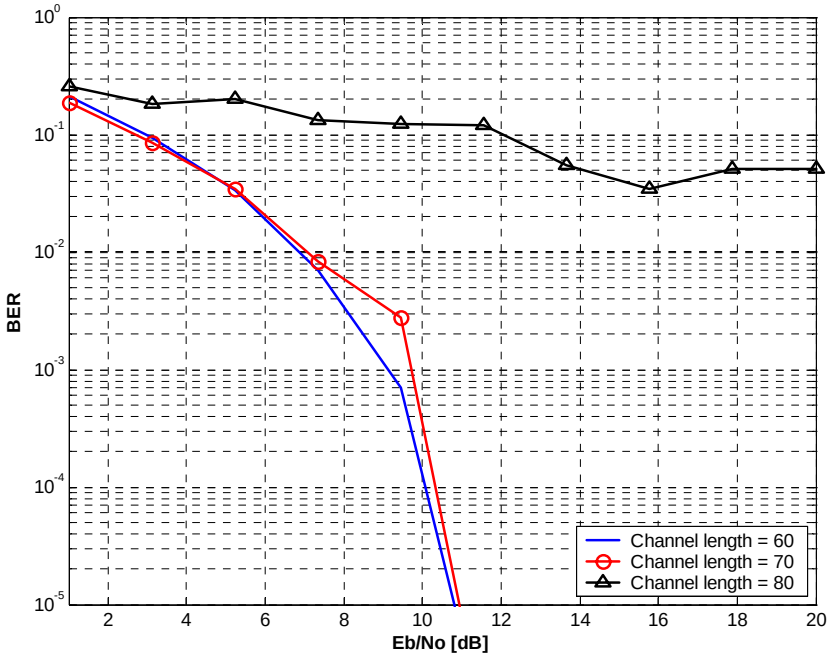

Figure 38. No Perturbation: BER in a Fading Channel with AWGN When Varying the Channel Length $\left(\text{packet size} = 72\,\text{bits}, R_b = 40\,\text{bps}, R_c = 2400\,\text{cps}\right)$.

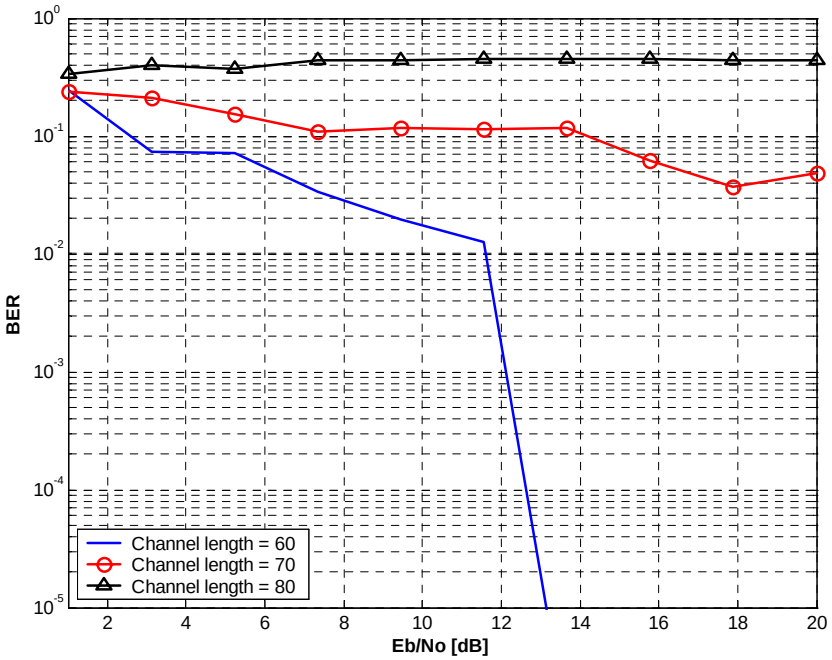

Figure 39. Interface Roughness Perturbation: BER in a Fading Channel with AWGN When Varying the Channel Length $\left(\text{packet size} = 72\,\text{bits}, R_b = 40\,\text{bps}, R_c = 2400\,\text{cps}\right)$.

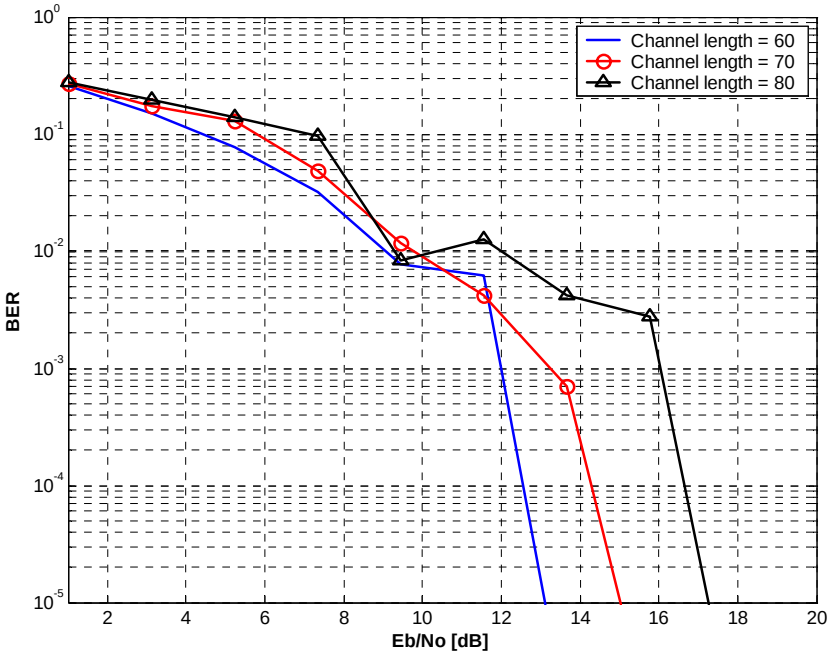

Figure 40. Internal Wave Perturbation: BER in a Fading Channel with AWGN When Varying the Channel Length $\left(\text{packet size} = 72\,\text{bits}, R_b = 40\,\text{bps}, R_c = 2400\,\text{cps}\right)$.

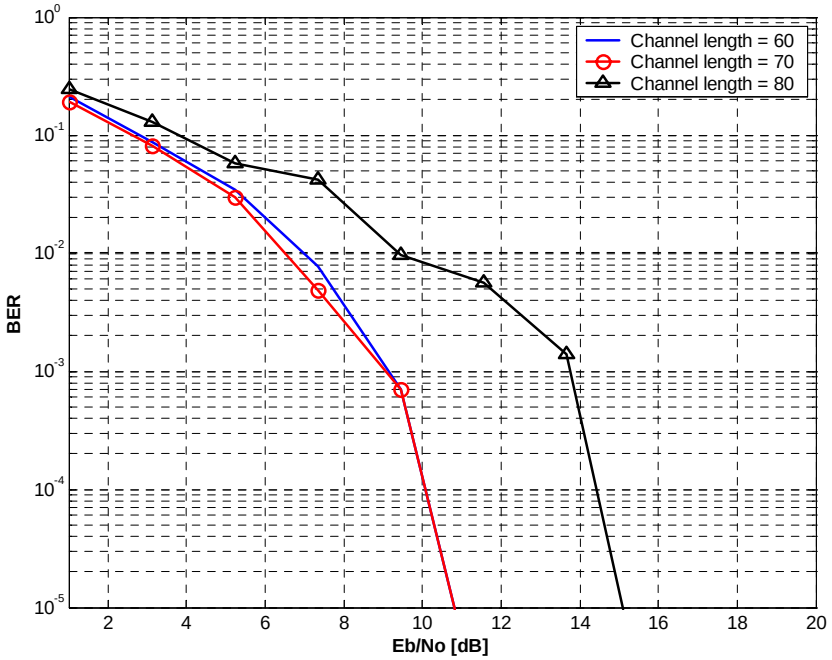

Figure 41. Bottom Volume Perturbation: BER in a Fading Channel with AWGN When Varying the Channel Length $\left(\text{packet size} = 72\,\text{bits}, R_b = 40\,\text{bps}, R_c = 2400\,\text{cps}\right)$.

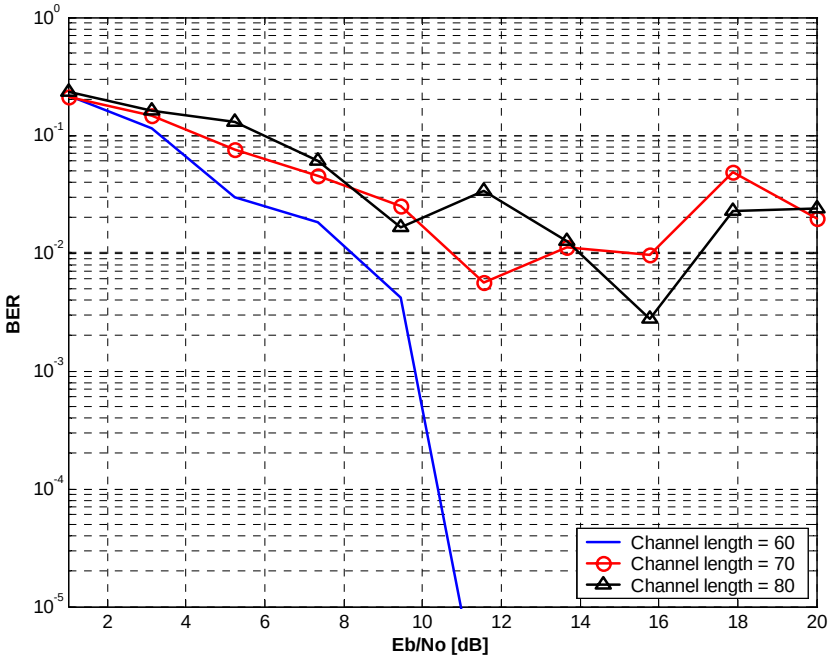

Figure 42. Turbulence Perturbation: BER in a Fading Channel with AWGN When Varying the Channel Length $\left(\text{packet size} = 72\,\text{bits}, R_b = 40\,\text{bps}, R_c = 2400\,\text{cps}\right)$.

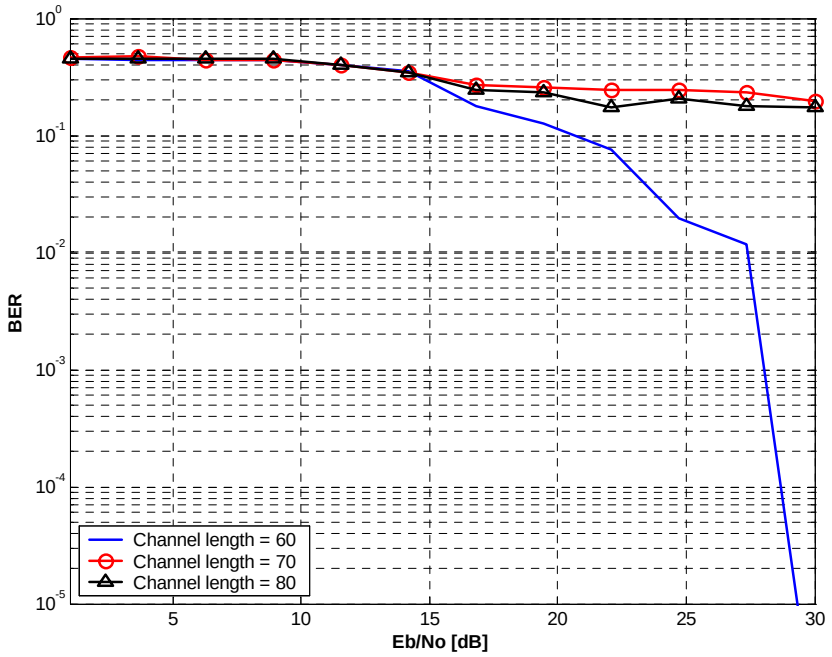

Figure 43. All Perturbations Combined (no Doppler): BER in a Fading Channel with AWGN When Varying the Channel Length $\left(\text{packet size} = 72\,\text{bits}, R_b = 40\,\text{bps}, R_c = 2400\,\text{cps}\right)$.

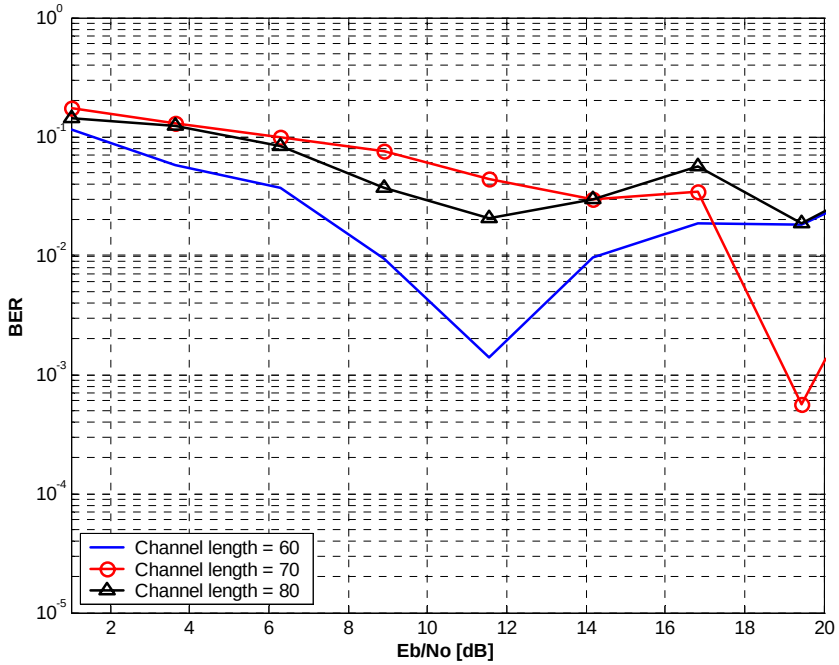

Figure 44. All Perturbations Combined (with Doppler): BER in a Fading Channel with AWGN When Varying the Channel Length $\left(\text{packet size} = 72\,\text{bits}, R_b = 40\,\text{bps}, R_c = 2400\,\text{cps}\right)$.

4. Performance in Multipath and Additive Color Noise

This is the same case as the previous one but colored noise was added instead of white. The equalization algorithm used in this case is described by Equations (5.11) through (5.15) where once again the CRV is used to estimate the signal and noise sub-spaces. The resulting BER for a channel length of 60 time samples is shown in Figure 45. The algorithm works well for all the individual perturbations but not when all of them are combined.

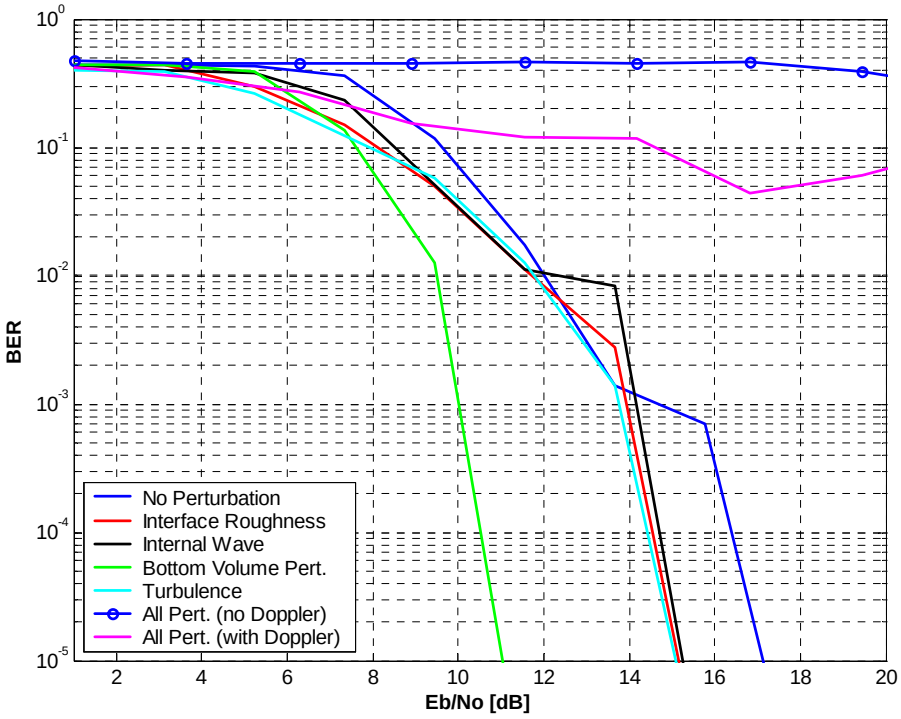

Figure 45. Performance in a Fading Channel with Additive Color Noise When Varying the Channel Length $\left(\text{packet size} = 72\,\text{bits}, R_b = 40\,\text{bps}, R_c = 2400\,\text{cps}\right)$.

5. Performance When Synchronization is Lost

When the PN-code in the transmitter and its replica in the receiver are synchronized (aligned in time), the received sequence is despread correctly. However due to the volatile underwater environment, this may not always be the case. When synchronization is lost, the dominant eigenvalues of the received covariance matrix are more than one. When at any given time $\underline{y}[n]$ contains two symbols (the current symbol $m[n]$ and a portion of the previous symbol $m[n-1]$), Equations (4.49) through (4.61) may be used for the recovery of the transmitted sequence. Therefore, in order to simulate the loss of synchronization, a random delay (of duration up to half a bit) was added to the received signal. The performance results of the algorithm are poor for all individual perturbations. For the case in which all perturbations are combined and subsequently the Doppler Effect is added the resulting BER's are shown in Figure 46.

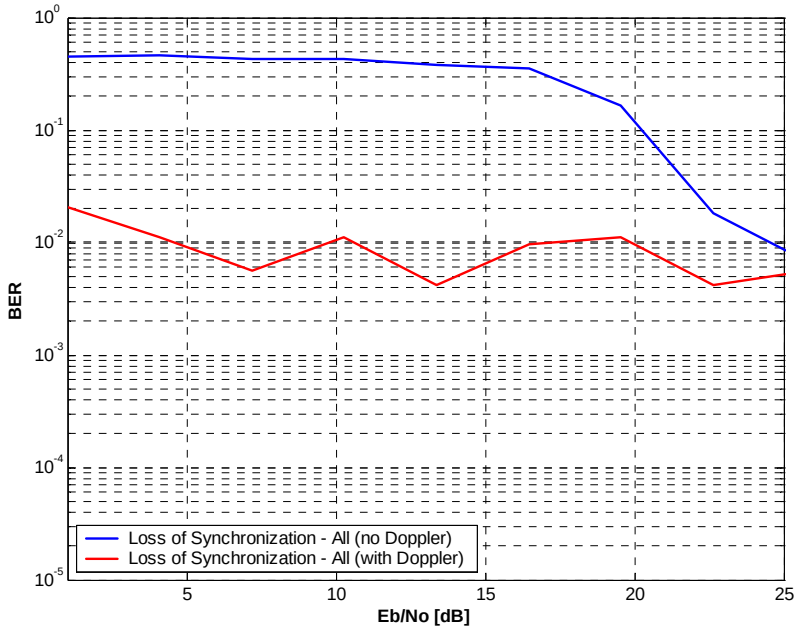

Figure 46. BER for the Case in which Synchronization of the PN-code is Lost.

6. Summary of Performance

In this chapter, the performance of the three different equalization algorithms, with the CRV used for the estimation of the signal and noise subspace, was examined under various design parameters and channel conditions.

When synchronization between the spreading sequence and its replica is achieved, the results of the proposed communication scheme are good for all the individual perturbations. For a fading channel with AWGN, a BER of 10^{-5} is possible for $E_b/N_o \geq 13\,\text{dB}$ whereas in a fading channel with additive color noise the same BER is achieved for $E_b/N_o \geq 17\,\text{dB}$.

Among the examined perturbations, the ones corresponding to the interface roughness and the turbulence have more impact on the performance of the communication scheme than the others. The small-scale fluctuations of sound speed within the water column caused by turbulence, and the scattering of the acoustic energy interacting with the rough bottom create a very challenging environment for underwater wireless communications due to the greater breakdown in the coherence of the propagating field.

The performance of the algorithms is not the desired one when all the perturbations are combined and Doppler is added to account for the source motion. But as already mentioned the modeled perturbations are rather strong, creating a shallow water environment more challenging than usual.

When synchronization is not achieved and the algorithm described earlier is used for the recovery of the information sequence, the results are very poor. As shown in Figure 46, a BER < 10^{-3} cannot be achieved even for $E_b/N_o = 25\,\text{dB}$. In the last chapter, the important findings of this thesis research are summarized and possible areas for follow-on work are presented.

VI. CONCLUSION

The primary purpose of this thesis was to evaluate the performance of the CRV updating algorithm developed in a previously proposed communication scheme intended for use in an underwater acoustic network operating in a shallow-water environment. The transmitter and receiver were implemented in MATLAB and the underwater channel was modeled using the Monterey-Miami Parabolic Equation model (MMPE). Various simulations were performed under different channel conditions (interface roughness, turbulence, etc.) and design parameters (packet size, bit rate).

A. FINDINGS

The CRV was proposed [9] as an alternative to the eigenvalue decomposition (EVD). The main advantage offered by the CRV is that it can be updated online at less computational cost compared to the EVD. As shown in Chapter IV, the estimates of the signal and noise subspace produced by the recursively updated algorithm are very accurate.

In order to examine the performance of the CRV, three distinct cases were simulated; a shallow-water fading channel with AWGN, a fading channel with .colored noise, and the case in which synchronization between the receiver and the transmitter were not achieved. Under all individual perturbations described in Chapter III, the equalization algorithm performed well and achieved a BER of $\leq 10^{-5}$ for values of $E_b/N_o \geq 13$ dB and $E_b/N_o \geq 17$ dB for white and colored noise, respectively.

Among the modeled perturbations, the ones corresponding to the interface roughness and turbulence were the most challenging for the proposed equalization algorithm. As mentioned in Chapter III, these perturbations cause the greater breakdown in the coherence of the propagating field resulting in a severe multipath environment.

On the other hand, when all perturbations were combined and the Doppler effect due to source motion was considered, the resulting bit-error-rates were not acceptable. For the case of no synchronization between the spreading and de-spreading sequence, the performance of the acquisition algorithm was also very poor.

B. FUTURE WORK

The channel impulse response was obtained by post processing the output of the MMPE acoustic propagation model, which represented the response of the simulated undersea environment. Therefore, the proposed communication scheme was tested under simulated conditions. It would be very interesting to evaluate its performance using real impulse responses obtained from experimental data in a variety of shallow-water channels.

In the current implementation, the desired subspaces (signal, noise) were estimated online with help of the recursively updated CRV algorithm. However, the equalization based on matched filter theory is still performed offline. More research on designing an effective communication scheme that could operate online is needed. This can be partially achieved by using training bits in each transmission or by using the first packet as a training transmission.

The CRV updating algorithm could be even more computationally efficient if the Lanczos bidiagonalization algorithm were used to compute the eigenvalues of the $\tilde{\Psi}$ matrix given by Equation (4.18).

There is a family of rank-revealing orthogonal decompositions, referred to as UTV decompositions [33], offered as alternatives to the SVD. These algorithms are able to provide all the necessary subspace information, and most importantly, can be updated online. A comparison of their performance, with the results obtained with the CRV decomposition, could lead to the design of a better signal processing algorithm for the recovery of the information signal in a multipath channel.

Finally, the performance of the proposed transmitter/receiver structure should be evaluated as part of a complete communication system with error correction coding and modulation.

APPENDIX. MATLAB CODE

The MATLAB source code used for the simulation of the communication system structure of Figure 31 and the implementation of the CRV updating algorithm is presented in this appendix. The overall controlling m-file is called *CRV_simulations.m.* All the other necessary functions (m-files) are called by running this file and are listed bellow.

CRV.m – This function implements the CRV updating algorithm described in Chapter IV. The output of this function is the recursively updated correlation matrix $\tilde{\Phi}$, and the matrices $\tilde{W}$ and $\tilde{V}$ needed for the initialization of the next step of the algorithm.

up.m – This function is used to upsample the message signal to the chipping rate so that the PN-code may be appended.

spredesp.m – This function is used to spread the binary message signal with the Gold code, or de-spread the received sequence with the replica of the Gold code used at the transmitter.

power_of_noise.m – This function calculates the noise power associated with a given E_b/N_o [dB], a bit length T_b [s], and a sampling period T_s [Hz]. The output is used to generate the white or colored noise corresponding to the noise power.

house.m – This function computes the Householder matrix *P* required to clear specific values of the received sequence (a required step in the CRV updating algorithm).

```
%**********************************
%          Overall Controlling Program
%               T/R   SIMULATION
% Note: all secondary variables and needed
% functions are called by executing this m-file
%
% developed by Pavlos Angelopoulos Mar 2004
% last modified 4/13
%**********************************
clear;
clc;
% Offer options for different simulations
disp('Choose one of the following options:');
disp('1.  White Noise only');
disp('2.  Color Noise only');
disp('3.  Fading Channel+AWGN');
disp('4.  Fading Channel+ACN');
disp('5.  Loss of Receiver Synchronization');

option=input(' ');
disp(' ');
if option==3|option==4;
   % Offer options for different perturbations
   disp('Choose one of the following perturbations:');
   disp('1.  Basic');
   disp('2.  Interface Roughness');
   disp('3.  Internal Wave');
   disp('4.  Volume');
   disp('5.  Turbulence');
   disp('6.  All Perturbations (without Doppler)');
   disp('7.  All Perturbations (with Doppler)');
   pert=input(' ');
   disp(' ');
   % Load the desired Impulse Response
   if      pert==1; load ImpRes_basic;
   elseif pert==2; load ImpRes_intrough;
   elseif pert==3; load ImpRes_intwave;
   elseif pert==4; load ImpRes_volume;
   elseif pert==5; load ImpRes_turbulence;
   elseif pert==6; load ImpRes_allpert;
   elseif pert==7; load ImpRes_doppler;
   end % end of "if pert"
   disp(' ');
end % end of "if option==3|option==4"

if option==3|option==4|option==5;
   disp('Choose the Channel Length:');
   channel_length=input(' ');
   disp(' ');
end % end of "if option==3|option==4|option==5"


disp('Choose # of Simulations:');
trials=input(' ');
disp(' ');
```

```
%-------------------------------------------------------------------------------
Rb = 40;                        % bit rate [bps]
Rc = 2400;                      % chip rate [cps]
BW=2*Rc;                        % bandwidth of spread signal
Tb = 1/Rb;                      % bit Period [s]
Tc = 1/Rc;                      % chip Period [s]
N = Rc/Rb;                      % processing gain
P=Tb/Tc;                        % Upsampling required to match chip sequence

load pn_code;                   % load PN chipping sequence "pn_code.mat"
packet_length = 72;                             % number of bits transmitted per packet
number_of_packets=1;        % number of packets transmitted
EbNo_dB=linspace(1,25,12); % determine Eb/No values to run simulations
disp('START');
disp(' ');
%-------------------------------------------------------------------------------
%-------------------------------------------------------------------------------
%                               TRANSMITTER
%-------------------------------------------------------------------------------
%-------------------------------------------------------------------------------
for simulation=1:trials;            % #of simulations
txt=sprintf('Simulation #%3d',simulation);
disp(txt);
tic;                                % starts clock to measure simulation run time
txt2=sprintf('Simulation #%3d',simulation);
h=waitbar(0,txt2);

for n=1:length(EbNo_dB);      % run simulation for each of the EbNo values
waitbar(n/length(EbNo_dB),h);

%-------------------------------------------------
% Generate the binary data sequence of +1,-1
%-------------------------------------------------
data_seq=sign(randn(1,number_of_packets*packet_length));

%-----------------------------------------------------------
% Upsample by P=Tb/Tc to match the chipping rate
%-----------------------------------------------------------
upsampled_data_seq=up(data_seq,P);
m=length(upsampled_data_seq);

%-----------------------------------------------------------
% Spread the data using with chipping sequence c[n]
%-----------------------------------------------------------
spreaded_signal=conv(upsampled_data_seq,pn_code(1:P));
spreaded_signal=spreaded_signal(1:m);

%-----------------------------------------
% Calculate the noise power associated
% with the simulated EbNo values
%-----------------------------------------
noise_power=power_of_noise(spreaded_signal,Tb,1/BW,EbNo_dB(n));
noise_std = sqrt(noise_power);  % noise standard deviation
```

```
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%                                   CHANNEL
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------

if option==1;
white_noise=randn(1,length(spreaded_signal));
received_seq= spreaded_signal + white_noise*noise_std;% Add White Noise
elseif option==2;
white_noise=randn(1,m);
color_noise=filter(1,[1 -0.9 0.4 -0.2 0.1 -0.1],white_noise);% filter white noise to produce color noise
received_seq= spreaded_signal + color_noise*noise_std;% Add Color Noise
elseif option==3;
spreaded_signal=conv(imp_res(1:channel_length),spreaded_signal);
white_noise=randn(1,length(spreaded_signal));
received_seq= spreaded_signal + white_noise*noise_std;% Add White Noise
elseif option==4;
spreaded_signal=conv(imp_res(1:channel_length),spreaded_signal);
white_noise=randn(1,length(spreaded_signal));
color_noise=filter(1,[1 -0.9 0.4 -0.2 0.1 -0.1],white_noise);
received_seq= spreaded_signal + color_noise*noise_std;% Add White Noise
else
load ImpRes_doppler;
spreaded_signal=conv(imp_res(1:channel_length),spreaded_signal);
white_noise=randn(1,length(spreaded_signal));
received_seq= spreaded_signal + white_noise*noise_std;% Add White Noise
end % end of "if option==1"

%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%                                   RECEIVER
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------

if option==5;
% Add random time t0 uniformly distributed between 0 and P
% to simulate the Loss of Synchronization between the codes
t0=round(rand*P/4)+1;
received_seq=received_seq(t0:length(received_seq));
end
%-----------------------------------------------------------------
% Despread the received sequence using a replica of c[n]
%-----------------------------------------------------------------
despreaded_signal=spredesp(received_seq(1:m),pn_code(1:P));
despreaded_signal=despreaded_signal(1:m);

%-----------------------------------------------
% Delay by z^-1 and Downsample by P
%-----------------------------------------------
Y=reshape(despreaded_signal,P,packet_length);
columns=size(Y,2);
```

```
%---------------------------------------
% Implement the CRV algorithm
%---------------------------------------

% Initialization
W=zeros(P,P);
V=eye(P);
r=0;  % Rank of matrix W (r=0)
for k=1:columns;
   [phi_crv,W,V,r]=CRV(Y(:,k),W,V,r); % call the "CRV.m" function
end

%-------------------------------------------------
% Compute the matched filter coefficients
%-------------------------------------------------

if option==1|option==3;
   [EigVec,EigVal] = eig(phi_crv);
   Opt_filter=EigVec(:,1);                % Eigenvector of the largest eigenvalue
elseif option==2|option==4;
   corrmatrix_color_noise=color_noise*color_noise';
   [EigVec_color,EigVal_color]=eig(corrmatrix_color_noise);
   R=inv(sqrtm(EigVal_color));
   R_sq_inv=EigVec_color*R*EigVec_color';
   R_mah=R_sq_inv*phi_crv*R_sq_inv;
   [EigVec_mah,EigVal_mah]=eig(R_mah);
   Opt_filter=R_sq_inv*EigVec_mah(:,1);
elseif option==5;
   [EigVec,EigVal] = eig(phi_crv);
   Opt_filter=EigVec(:,1);                % Eigenvector of the largest eigenvalue
   EigVal=diag(EigVal);
   if EigVal(2)>0.2*EigVal(1);          % Assume there is no synchronization
   lamda=EigVec(:,1:2)'*Y;
   E=[lamda(:,1:length(lamda)-1); lamda(:,2:length(lamda))];
   R_lamda=E*E';
   [Ve, De]=eig(R_lamda);
   lam_e=diag(De);
   I=find(lam_e==min(abs(lam_e)));
   ke=Ve(:,I);
   k1=ke(1:2);
   k2=-ke(3:4);
   end
end

%--------------------------------
% Filter the received signal
%--------------------------------
if option==5 & EigVal(2)>0.1*EigVal(1);
   decoded_seq=sign([k1',k2']*E);
else
   decoded_seq=sign(Opt_filter'*Y) ;
end;
```

```
%----------------------
% Calculate  Errors
%----------------------
if option==5 & EigVal(2)>0.1*EigVal(1);
diff=length(decoded_seq)-length(data_seq);
   if diff>0
      err=length(find(data_seq-decoded_seq(diff:length(data_seq)+diff-1)~=0));
   elseif diff<0
      diff=abs(diff);
      err=length(find(data_seq(diff:length(decoded_seq)+diff-1)-decoded_seq~=0));
   else
      err=sum(abs(data_seq-decoded_seq))/2;   % # of errors
   end % end of "if diff>0"
else
err=sum(abs(data_seq-decoded_seq))/2;   % # of errors
end % end of "if option==5 & EigVal(2)>0.1*EigVal(1)"
if err>packet_length/2;
  channel_errors(n)=packet_length-err;
else
  channel_errors(n)=err;
end % end of "if err>packet_length/2"

end               % end of n loop
close(h);
toc;
summary(simulation,:)=channel_errors./packet_length;

end               % end of simulations loop
avg_errors=mean(summary);

result=[EbNo_dB; avg_errors];
%--------------------------------------------------------------------
%                      Plot Simulation Results
%--------------------------------------------------------------------
result(2,find(result(2,:)==0))=10^-6;
figure;
semilogy(result(1,:),result(2,:),'-','MarkerSize',8);
grid on;
ylim([10^-5 1]); xlim([1 30]);
ylabel('BER','fontsize',13);
xlabel('Eb/No (dB)','fontsize',13);
legend(sprintf('Packet size: %3.0f bits',packet_length));

disp('STOP');
% offer option to save the simulation results
% for post processing (plots)
 savedata=input('Save Simulation Results? (y or n) ','s');
 disp(' ');
  if savedata == 'y' | savedata == 'Y'
    fileout=input('Enter Output filename (without extension): ','s');
    eval(['save ' fileout ' result trials packet_length Rc;']);
  end
  if savedata == 'n' | savedata == 'N'
  end
break
%-------------------------END of Code------------------------------
```

```
function [phi_crv,W,V,r]=CRV(signal,W,V,r)

%************************************************
%                    CRV UPDATED ALGORITHM
% "signal" is the new signal used to update the correlation matrix
% "W" and "V" are the matrices needed to initialize the algorithm
% "r" is the rank of W
% "phi_crv" is the esimate of the correlation matrix
%
%  developed by Pavlos Angelopoulos Feb 2004
%  last modified 3/28
%************************************************
a=1;  % Fading Factor
P=length(signal);
W=a*W;
z=V'*signal;

   if r~=0;
      if r==P;
         W=W+z*z';
      else
        P_hat=house(z(r+1:end));                    % Householder vector: keep r+1 element and zero the rest
        P_diag=[eye(r) zeros(r,P-r);zeros(P-r,r) P_hat];   % P=diag(Ir,P_hat)
        W=P_diag'*(W+z*z')*P_diag;                  % update matrix W
        V=V*P_diag;                                 % update matrix V
        warning off;
        eigenv=abs(eigs(W(1:r+1,1:r+1),r+1));       % compute eigenvalues of (r+1)x(r+1) block of W
        if eigenv(r+1)>0.1*eigenv(r);               % Threshold to check rank change
        r=r+1;
        end
      end
      phi_crv=V*W*V';
   else
      W=W+z*z';
      phi_crv=V*W*V';
      r=rank(W);
   end
%---------------END of function CRV------------------



function y=up(x,M)

%************************************************
%  Upsample: A function to upsample a signal x by an integer M
%
%  developed by Pavlos Angelopoulos Nov 2003
%  last modified 3/28
%************************************************
n=1:length(x);
y(n*M)=x;
y=[y(M:end) zeros(1,M-1)];
%----------------End of function "up"--------------------------
```

```
function noise_power=power_of_noise(x,Tb,Ts,EbNo_dB)

%*****************************************
% Noise Power
% A function to calculate the noise power associated with a
% given Eb/No (dB), bit length Tb, and sampling period Ts
%
% developed by Pavlos Angelopoulos Nov 2003
% last modified 11/03
%*****************************************
signal_power=sum(x.^2)/length(x);
EbNo=10^(EbNo_dB/10);
Eb=signal_power*Tb;                % energy per bit
No=Eb/EbNo;
sigma_No=sqrt(No/(2*Ts));      % std deviation of noise
noise_power=sigma_No^2;
%------------End of function "power_of_noise"--------------



function P=house(x)

%*********************************************
% A function to compute the householder matrix P such that
% transformed_x=P*x is a vector whose elements except the
% first one are equal to 0.
% x must be defined as a column vector
%
% developed by P.Angelopoulos Nov 2003
% last modified 11/24
%*********************************************
k=length(x);
e1=[1,zeros(1,k-1)]';
v=x+sign(x(1))*norm(x,2)*e1;
P=eye(k)-2*v*v'./(v'*v);
%-------------------End of function "house"----------------------



function data=spredesp(x,code)

%*****************************************************
% A function to spread (chip) a binary signal x with gold code "code"
% or de-spread the received signal by a replica of the "code"
%
% developed by P.Angelopoulos Oct 2003
% last modified 10/23
%*****************************************************
k=length(x);
m=length(code);
repeated_code=repmat(code,1,ceil(k/m));
data=x.*repeated_code;
%---------------------End of function "spredesp"-----------------------
```

LIST OF REFERENCES

[1] P. Duke, "Direct-Sequence Spread-Spectrum Modulation for Utility Packet Transmission in Underwater Acoustic Communication Networks", Master's Thesis, Naval Postgraduate School, Monterey, California, 2002.

[2] G. N. Pelekanos, "Performance of Acoustic Spread-Spectrum Signaling in Simulated Ocean Channels", Master's Thesis, Naval Postgraduate School, Monterey, California, 2003.

[3] J. A. Rice, R. K. Creber, C. L. Fletcher, P .A. Baxley, D .E. Rogers, and D. C. Davsion, "Seaweb Underwater Acoustic Nets," *Space and Naval Warfare Systems Center San Diego Biennial Review 2001*, pp. 234-243, 2001.

[4] Joe Rice, Bob Creber, Chris Fletcher, Paul Baxley, Ken Rogers, Keyko McDonald, Dave Rees, Michael Wolf, Steve Merriam, Rami Mechio, John Proakis, Ken Scussel, Dave Porta, John Baker, Jim Hardiman, and Dale Green, "Evolution of Seaweb Underwater Acoustic Networking", *IEEE*, pp. 2007-2017, 2000.

[5] M. B. Porter, "Ocean Acoustics Library," [http://oalib.saic.com], last accessed February 2004.

[6] NATO SACLANT, "Acoustic Models", [http://www.saclantc.nato.int/frameset-ch6.html], last accessed February 2004.

[7] K. B. Smith, "Convergence, stability, and variability of shallow water acoustic predictions using a split-step Fourier parabolic equation model", *J. Comp. Acoust.*, 9, pp. 243-285, 2000.

[8] J. G. Proakis, *Digital Communications,* 4th ed., pp. 726-727, McGraw-Hill, New York, 2001.

[9] E. S. Baker and R. D. DeGroat, "A Correlation-Based Subspace Tracking Algorithm", *IEEE Transactions on Signal Processing,* Vol. 46, pp. 3112-3116, 1998.

[10] Lawrence E. Kinsler, Austin R. Frey, Alan B. Coppers, and James V. Sanders, "Underwater Acoustics," in *Fundamentals of Acoustics,* 4th ed., pp. 435-470, John Wiley & Sons, New York, 2000.

[11] Finn B. Jensen, William A. Kuperman, Michael B. Porter, and Henrik Schmidt, *Computational Ocean Acoustics,* AIP Press, New York, 1994.

[12] Robert J. Urick, *Principles of Underwater Sound,* 3rd ed., McGraw-Hill, New York, 1983.

[13] L. M. Brekhovskikh, and Yu. P. Lysanov, *Fundamentals of Ocean Acoustics,* 3rd ed., Springer-Verlag, New York, 2002.

[14] Robert J. Urick, *Ambient Noise in the Sea,* Peninsula Publishing, Los Altos, California, 1986.

[15] Gordon M. Wenz, "Acoustic Ambient Noise in the Ocean: Spectra and Sources," *Journal of the Acoustical Society of America*, Vol. 34, No. 12, pp. 1936-1956, 1962.

[16] Rodney F. W. Coates, "Noise and Reverberation," in *Underwater Acoustic Systems*, pp. 90-94, John Wiley & Sons, New York, 1989.

[17] www.benthos.com/pdf/Telesonar%20Transducers.pdf, last accessed March 2004.

[18] Theodore S. Rappaport, *Wireless Communications: Principles and Practices,* 2nd ed., Prentice Hall, Upper Saddle River, New Jersey, 2002.

[19] F. D. Tappert, "The parabolic approximation method" (Chapter V). *Lecture Notes in Physics,* Vol. 70, Wave Propagation and Underwater Acoustics, eds. J. B. Keller and J. S. Papadakis, Springler-Verlag, New York, 1977.

[20] R. H. Hardin and F. D. Tappert, "Applications of the split-step Fourier method to the numerical solution of nonlinear and variable coefficient wave equations," *SIAM Rev.*, Vol. 15, pg. 423, 1973.

[21] J. A. Goff and T. H. Jordan, "Stochastic modeling of seafloor morphology: Inversion os Sea Beam data for second-order statistics," *J. Geophys. Res*., Vol. 93, pp. 13589-13609, 1988.

[22] Robert M. Hill, "Model-Data Comparison of Shallow Water Acoustic Reverberation in the East China Sea," Master's Thesis, Naval Postgraduate School, Monterey, California, 2003.

[23] T. Yamamoto, "Velocity variabilities and other physical properties of marine sediments measured by croswell acoustic tomography," *Journal of the Acoustical Society of America*, Vol. 98, pp. 2235-2248, 1995.

[24] V. I. Tatarski, *Wave Propagation in a Turbulent Medium*, Dover Publications, London, 1961.

[25] T. F. Duda, S. M. Flatte', and D. B. Creamer, "Modelling Meter-Scale Acoustic Intensity Fluctuations From Oceanic Fine Structure and Microstructure," *Journal of Geophysical Research*, Vol. 93, No. C5, pp. 5130-5142, 1988.

[26] F. S. Henyey, D. Rouseff, J. M. Grochocinski, S. A. Reynolds, K. L. Williams, and T. E. Ewart, "Effects of Internal Waves and Turbulence on a Horizontal Aperture Sonar," *IEEE Journal of Oceanic Engineering*, Vol. 22, No. 2, pp. 270-280, 1997.

[27] A. Tolstoy, K. Smith, and N. Maltsev, "The SWAM'99 Workshop – an Overview," *Journal of Computational Acoustics*, Vol. 9, No. 1, pp. 1-16, 2001.

[28] K. B. Smith, "Computing the Influence of Doppler Due to Source/Receiver Motion in a Parabolic Equation Model," *Journal of Computational Acoustics*, Vol. 10, No. 3, pp. 295-309, 2002.

[29] G. W. Stewart, "An Updating Algorithm for Subspace Tracking," *IEEE Transactions on Signal Processing*, Vol. 40, No. 6, pp. 1535-1541,1992.

[30] G. H. Golub, and C. F. VanLoan, *Matrix Computations*, 3rd ed., John Hopkins University Press, Baltimore, 1996.

[31] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press Inc., London, 1973.

[32] Charles W. Therrien, *Discrete Random Signals and Statistical Signal Processing,* Prentice-Hall, Upper Saddle River, New Jersey, 1992.

[33] G. W. Stewart, *Matrix Algorithms. Volume 1: Basic Decompositions*, SIAM, Philadelphia, 1998.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman, Code EC
   Electrical and Computer Engineering Department
   Naval Postgraduate School
   Monterey, California

4. Chairman, Code EA
   Engineering Acoustics Academic Committee
   Naval Postgraduate School
   Monterey, California

5. Prof. Roberto Cristi, Code EC/Cx
   Electrical and Computer Engineering Department
   Naval Postgraduate School
   Monterey, California

6. Mr. Joe Rice, Code PH/Rj
   Physics Department
   Naval Post Graduate School
   Monterey, California